

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Imagerie médicale

#### Acquisition multimodale en imagerie médicale et application à l'endoscopie

Leclère, Maylis

*Award date:*  
2007

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Univrsitaires Notre-Dame de la Paix, Namur  
Institut d'Informatique

Année Académique 2006–2007

Imagerie Médicale

Acquisition multimodale en imagerie médicale  
et application à l'endoscopie

Leclere Mailys

Mémoire présenté en vue de l'obtention du grade de Maître en Informatique.



## **Résumé**

De nos jours, chaque discipline médicale possède ses propres techniques d'imagerie médicale, elles permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour de nombreuses maladies. La détection avec plus de rapidité et de précision des causes d'un problème de santé améliore les décisions prises concernant les thérapies à envisager et facilite le recours à la chirurgie améliorant ainsi les chances de guérison. L'endoscopie est un cas particulier d'imagerie médicale, elle permet d'explorer, à l'aide d'images vidéos, l'intérieur des cavités du corps comme par exemple le colon, dans le but de déceler des anomalies. Ce mémoire propose, à partir des besoins exprimés par le service d'endoscopie de l'hôpital de Mont-Godinne, un outil informatique polyvalent et orienté-objet appelé "endoscope virtuel" qui permet l'acquisition, la visualisation et le stockage de la vidéo provenant d'un appareil de capture connecté sur le port usb ou firewire d'un ordinateur. Une autre application est aussi présentée, elle permet d'intégrer différents outils d'imageries dans une seule et même interface afin de partager les fonctionnalités utiles à chaque outil. Cette plate-forme d'intégration permet aussi de faciliter la gestion et le développement des applications d'imageries.

**Mots-clefs:** Imagerie médicale, Orienté-objet, Endoscopie, JMF, JAVA, Acquisition vidéo, Acquisition multimodale.

## **Abstract**

Today, every department of a hospital has his own medical imagery technics, these allows a better diagnostic but also offers new hopes of therapy for a lot of deseases. A faster and more accurate detection of the causes of a health problems makes better the decisions concerning the therapies to choose and makes easier the chirurgicals operations to boost the cure rating. This memory proposes in accord to the needs expressed by the endoscopy service of the Mont-Godinne Hospital, a general-purpose and object-oriented informatic tool called "endoscope virtuel", it allows the acquisition, the visualisation and the storage of video that comes from a capture device connected to the usb or firewire port of a computer. Another application is also presented; it allows to integrate differents medical imagery tools in the goal to provide the usefull functionalities for each of them. This integration platform gives also an easier management and development of imagery applications.

**Key-words:** Medical imagery, Object-oriented, Endoscopy, JMF, JAVA, Video acquisition, Multimodal acquisition.



*Remerciements :*

*Je tiens à remercier le professeur Jean-Pol Leclercq, promoteur de ce mémoire, pour ses relectures et ses conseils sur la réalisation de ce travail.*

*Je remercie également Monsieur Hubert Meurisse, Maître de stage et co-promoteur , pour m'avoir proposé un sujet de stage qui à donné lieu à ce mémoire. Je lui suis également reconnaissant pour son aide sur la réalisation de ce mémoire.*

*Je remercie ma mère pour son aide quand à la rédaction de ce mémoire ainsi que pour ses relectures.*

*Enfin je remercie Cédric Peeters, stagiaire à l'hôpital Mont-Godinne, pour sa bonne humeur et sa sympathie.*



# Table des matières

Introduction.....	6
Partie I Etat de l'art.....	8
Chapitre 1 L'informatique médicale et l'hôpital de Mont-Godinne.....	8
1.1 Introduction.....	8
1.2 La télémedecine.....	8
1.2.1 Définition.....	8
1.2.2 L'application Microscope Virtuel.....	8
1.3 Techniques d'imagerie médicale et l'endoscopie.....	9
1.3.1 Définition .....	9
1.3.2 Endoscopie.....	9
1.3.3 Apport de l'informatique à l'endoscopie.....	10
1.4 Inter-opérabilité, la norme DICOM.....	13
1.4.1 Définition.....	13
1.4.2 L'application Galerie.....	14
1.5 La nécessité d'une application d'endoscopie.....	14
1.6 La nécessité d'une application d'intégration.....	14
1.7 Conclusion.....	14
Partie II Objectifs et Choix.....	16
Introduction.....	16
Chapitre 2 Une plate-forme d'intégration.....	16
2.1 Présentation du projet.....	16
2.2 Une interface ergonomique et intuitive.....	16
2.3 Ajouter et rassembler des fonctionnalités communes.....	17
2.4 Une conception qui permet flexibilité et maintenabilité.....	17
Chapitre 3 Choix effectués pour l'Interface Commune.....	17
3.1 Re-conception et décomposition de l' application 'Galerie' .....	17
Chapitre 4 Un Endoscope Virtuel.....	18
4.1 Présentation du projet.....	18
4.2 Objectifs 2006-2007 .....	18
4.3 Une modélisation qui permet maintenabilité et simplicité.....	18



Chapitre 5 Choix effectués pour l' Endoscope Virtuel.....	18
5.1 Le matériel de test.....	18
5.2 L' API JMFF.....	19
5.3 L' enregistrement en MPEG2 MP@ML.....	19
Chapitre 6 Introduction à JMFF.....	19
6.1 Introduction.....	19
6.2 Possibilités.....	19
6.3 Architecture.....	20
6.3.1 DataSource.....	20
6.3.2 CaptureDevice.....	20
6.3.3 Player.....	21
6.3.4 Processor.....	21
6.3.5 DataSink .....	21
6.3.6 Format.....	22
6.3.7 Manager.....	22
6.4 Installation et enregistrement d'un périphérique.....	22
6.4.1 Installation.....	22
6.4.2 Enregistrement d'un appareil de lecture.....	22
 Partie III Résultats.....	 24
Introduction.....	24
Chapitre 7 L'interface commune.....	24
7.1 Introduction.....	24
7.2 Diagramme de classe général.....	24
7.3 Explication du diagramme.....	25
7.3.1 Les classes 'Fenêtre' et 'Onglet' .....	25
7.3.2 Le concept de 'Outil' .....	25
7.3.3 Le concept de 'composant partagé' .....	26
7.4 Le concept de 'Source'.....	28
7.4.1 Explication du concept.....	28
7.4.2 Le diagramme de classe.....	28
7.4.3 Explication des classes.....	28
7.5 Screenshots de l'application.....	29
7.5.1 Vue d'ensemble avec l'outil 'Galerie' .....	29
7.5.2 L'onglet 'Panel Envois'.....	30
7.5.3 Autre exemple, le Microscope Virtuel.....	31
7.6 Diagrammes de séquence.....	32
7.6.1 Diagramme de séquence pour accéder aux informations contextuelles.....	32
7.6.2 Explication du diagramme.....	32
7.6.3 Diagramme de séquence pour envoyer un fichier dans le panier.....	33
7.6.4 Explication du diagramme.....	33
7.7 Conclusion.....	33

Chapitre 8 Le prototype de lecteur vidéo.....	34
8.1 La conception, le diagramme de classe.....	34
8.2 Explication des classes.....	34
8.3 Screenshots de l'application.....	35
8.3.1 Le lecteur vidéo intégré dans l' Interface, avec la webcam en entrée.....	36
8.3.2 Le panier avec un fichier de capture vidéo.....	37
8.3.3 Le lecteur avec en entrée le système Pinnacle et le magnétoscope.....	38
8.3.4 Lecture d'un fichier stocké sur le disque dur.....	38
8.4 Conclusion.....	39
 Chapitre 9 Etude de la norme MPEG-2 .....	40
9.1 Introduction.....	40
9.2 MPEG-2 Présentation.....	40
9.2.1 Présentation du MPEG-2.....	40
9.2.2 La notion de codec41	
9.3 La norme MPEG-2.....	41
9.4 Les fonctionnalités.....	42
9.5 Profile and Level.....	42
9.6 Le flux de données.....	44
9.6.1 Hiérarchie du flux vidéo MPEG-2 .....	44
9.6.2 Picture Types.....	46
9.7 L'algorithme de codage.....	47
9.7.1 Codage des I-picture.....	47
9.7.2 Explication des différents étapes.....	47
9.7.3 Codage des P-picture .....	51
9.7.4 Codage des B-picture.....	52
9.8 Conclusion.....	53
 Partie IV Discussion.....	54
introduction.....	54
10.1 Interface Commune, discussion de la solution.....	54
10.1.1 Une interface ergonomique et intuitive.....	54
10.1.2 Ajouter et rassembler des fonctionnalités commune.....	54
10.1.3 Une conception qui permet flexibilité et maintenabilité.....	54
10.1.4 Remarque.....	55
10.2 Interface Commune, perspectives.....	55
10.2.1 La sélection des outils.....	55
10.3 Endoscope Virtuel, discussion de la solution.....	55
10.3.1 Les Fonctionnalités de bases.....	55
10.3.2 Affichage du flux vidéo.....	56
10.3.3 Autres remarques sur l'application actuelle.....	56
10.4 Endoscope Virtuel, perspectives.....	56
10.4.1 L'ajout d'effets visuels.....	56
10.4.2 Les bibliothèques open-source.....	57

V Conclusion.....	59
Bibliographie.....	60
Annexe I.....	61
Annexe II.....	64

# Table des figures

## Partie I

1) Fibroscope.....	10
2 ) Reconstitution 3D, à droite l'image de départ.....	11
3) Reconstitution 3D num 2, à droite l'image.....	11
4 ) Système d'aide pour opération.....	12
5 ) Segmentation couleur.....	13
6 ) Segmentation textures.....	13
7) Mise en évidence du lumen.....	13

## Partie II

8) JMFregistry .....	23
----------------------	----

## Partie III

9) Diagramme de classe général.....	25
10) Diagramme de classe 'Source' .....	28
11) L'outil Galerie.....	29
12) L'outil PanelEnvois.....	30
13) Le Microscope Virtuel.....	31
14) Diagramme de séquence num 1 .....	32
15) Diagramme de séquence num 2.....	33
16) Diagramme de classe Endoscope Virtuel.....	34
17) Lecteur vidéo avec webcam.....	36
18) Panier avec lecture d'un fichier audio.....	37
19) Lecteur avec magnétoscope.....	38
20) Lecteur avec fichier.....	38
21) Combinaison Levels et Profiles.....	43
22) Les différents Levels de la norme MPEG-2.....	43
23) Les différents Profiles de la norme MPEG-2.....	44
24) La hiérarchie du flux MPEG-2.....	44
25) Macroblock.....	46
26) Séquence d'images MPEG-2.....	46
27) Algorithme d'encodage MPEG-2.....	47
28) Algorithme P-picture.....	51
29) Algorithme B-picture.....	52

## Partie IV

30) JMF-effet-exemple 2.....	57
31) JMF-effet-exemple 1.....	57

# INTRODUCTION

De tout temps, la médecine a su tirer parti des technologies de son époque afin de développer des outils qui lui permettent de parfaire son art de la diagnostique ou d'appliquer de nouvelles thérapies. Ainsi l'accaparement du paradigme informatique par la médecine a donné naissance à l'informatique médicale, c'est à dire l'application de technologies issues de l'informatique au domaine médical. Les nouveaux outils développés servent non seulement à l'archivage d'une quantité toujours plus importante d'informations, à la communication inter-services ou hôpitaux mais surtout, c'est à l'imagerie médicale que l'informatique apporte sa plus grande aide de diagnostique.

Entre autre, ce croisement entre les nouvelles technologies de l'information et de la communication avec les techniques médicales a donné naissance à de nouvelles façons d'exercer la médecine au quotidien. Nous pouvons citer comme exemple la télémedecine, qui permet à un scientifique d'effectuer son travail à partir d'un poste éloigné géographiquement de son laboratoire ; l'échange d'informations entre des hôpitaux ou centres de recherches ; ou encore, le développement de nouvelles technologies d'imageries , servant notamment d'aide au diagnostic et, de support lors d'opérations chirurgicales complexes.

Un hôpital moderne est donc à travers les différents outils informatiques qui compose son système d'information, producteur d'une multitude de fichiers qu'ils faut pouvoir gérer, stocker, distribuer ou encore transformer. Ces fichiers concernent aussi bien de simples informations textuelles que des images, des vidéos, des graphiques,etc.

L'évolution rapide des technologies de l'information et de la communication fait, de part les possibilités qu'elles laissent entrevoir et la complexité qu'elles engendrent, constamment évoluer les besoins en logiciels informatiques des différents acteurs du monde de la santé.

Ce travail basé sur le stage 2006-2007 effectué dans le service de recherche informatique dirigé par Monsieur Hubert Meurisse propose de se focaliser plus particulièrement sur les besoins exprimés par le service d'endoscopie de l'hôpital de Mont-Godinne , mais aussi, sur les difficultés rencontrées par le service de recherche informatique dans la gestion des applications d'imagerie médicale toujours plus nombreuses .

Ce mémoire a donc pour objectif de faciliter et d'améliorer le travail fournit par les services ayant recours à l'équipe de recherche informatique et plus particulièrement au service d'endoscopie.

La première partie, nommée "Etat de l'art" présente brièvement les trois domaines de l'informatique médicale exposés précédemment, la télémedecine, l'inter-opérabilité et les techniques d'imageries médicales dans l'endoscopie. Elle expose ensuite l'état actuel des logiciels développés dans ce cadre et la nécessité de lancer deux nouveaux projets pour répondre aux lacunes des applications existantes .

La deuxième partie "Objectifs et Choix" détaille les objectifs à atteindre pour cette année 2006-2007 ainsi que les choix entrepris pour mener à bien les nouveaux projets débutés cette année.

La troisième partie "Résultats" présente les applications développées à travers l'architecture interne exprimée en diagramme de classes mais aussi une série de screenshots qui illustrent plus

concrètement les solutions proposées.

Enfin, la quatrième partie tentera de vérifier si les solutions informatiques répondent correctement aux besoins exprimés par les scientifiques de l'hôpital. Des idées pour l'évolution des applications seront ensuite présentées afin de permettre aux projets d'être menés à bien sur le long terme.

# **PARTIE I Etat de l'art**

## **Chapitre 1 L'informatique médicale et l'hôpital de Mont-Godinne**

### **1.1 Introduction**

Nous allons, dans ce chapitre , définir brièvement les trois aspects de l'informatique médicale que sont la télémedecine, la communication inter-système et les techniques d'imageries endoscopiques . Des exemples d'applications seront présentées . Nous en tirerons ensuite la nécessité pour l'hôpital de Mont-Godinne de développer de nouveaux logiciels qui lui permettrons d' accroire la qualité et la rapidité du service médical fournit à ses patients. Ces nouvelles exigences informatiques donneront naissance à deux nouveaux projet dont ce travail est le point de départ.

### **1.2 La télémedecine**

#### **1.2.1 Définition**

Le domaine d'application de la télémedecine est très vaste. Il englobe une série de pratiques toutes aussi différentes les unes des autres mais dont l'objectif final est le même à savoir : la prestation à distance de soins médicaux.

Parmi ces applications, la télémicroscopie est une discipline qui permet à un médecin de visualiser à distance une lame porte-objets afin de rendre un diagnostic. La problématique d'une telle application réside dans la capacité à offrir aux médecins, habitués à travailler avec un microscope conventionnel, une application leur permettant de prester leur savoir-faire à distance de la manière la plus optimale possible.

Définition inspirée de la référence [JADC].

#### **1.2.2 L'application Microscope Virtuel**

Depuis plusieurs années, des mémorants se sont succédés au sein du laboratoire de cytologie de Mont-Godinne afin de développer une application de télémicroscopie. Ce projet est toujours en cours de développement.

L'objectif principal de cette application de visualisation est de permettre à un médecin extérieur au laboratoire, de visualiser des mégaimages représentant des lames virtuelles précédemment capturées et stockées sur un serveur d'images.

Actuellement, cette application permet de visualiser à distance des images capturées à l'aide d'un microscope, mais aussi , de zoomer, dézoomer et générer automatiquement une série d'images représentant des anomalies dans l'image d'origine. Cette application est toujours en cours de développement et de nouvelles fonctionnalités seront ajoutées cette années. Pour plus d'information sur l'état courant du projet, voir le mémoire de Cédric Peeters ".....".

## 1.3 Techniques d'imagerie médicale et l'endoscopie

### 1.3.1 Définition

L'imagerie médicale est certainement l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Ces récentes découvertes permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour de nombreuses maladies. Cancer, épilepsie... l'identification précise de la lésion (anomalie affectant un organe interne) facilite déjà le recours à la chirurgie, seule solution thérapeutique pour certains malades. De telles techniques permettent également de mieux comprendre le fonctionnement de certains organes encore mystérieux, comme le cerveau.

En voici une définition:

*L'imagerie médicale est le procédé par lequel un médecin peut examiner l'intérieur du corps d'un patient sans l'opérer. L'imagerie médicale peut être utilisée d'une part à des fins cliniques à la recherche d'un diagnostic ou pour le traitement d'un grand nombre de pathologies mais également pour la recherche dans le but d'étudier la physiologie des êtres vivants. [wik]*

Toutes les grandes disciplines médicales possèdent aujourd'hui ses propres techniques d'imagerie. Parmi-elles, nous pouvons citer la radiographie, qui permet de réaliser des clichés à l'aide de rayons X des structures internes d'un patient, les techniques modernes alliées à l'informatiques ont donné naissance à la radiographie informatisée(CR) et la radiographie numérique direct(DR). Une autre technique est l'imagerie par résonance magnétique nucléaire(IRM), elle utilise un gros aimant et analyse la réaction des différents tissus du corps à des champs magnétiques. Les données recueillies sont ensuite traitées informatiquement et la zone étudiée peut être restituée en deux ou trois dimensions.

Un autre domaine, qui nous intéresse plus particulièrement dans le cadre de ce mémoire est l'endoscopie.

### 1.3.2 Endoscopie

*L'endoscopie est une méthode d'exploration et d'imagerie médicale qui permet de visualiser l'intérieur (endo en grec) des conduits (tube digestif) et des cavités du corps et à les éclairer à l'aide d'une lumière extérieure dont les rayons étaient autrefois réfléchis par les parois de l'appareil (e. à lumière externe de Désormaux) ou bien actuellement, soit à l'aide d'un endoscope soit par l'ingestion d'une capsule de la taille d'une gélule.*

*L'endoscope est composé d'un tube optique muni d'un système d'éclairage et de petites caméras vidéos qui peuvent ainsi retransmettre l'image sur un écran. [WIK]*

Voici une image illustrant un fibroscope (endoscope particulier) :





*Illustration 1: Fibroscope*

*Un fibroscope est une variété d'endoscope conduisant les rayons lumineux par un faisceau de fibres de verre souples. Le fibroscope permet d'explorer de façon très complète (par vision directe, photographie, cinématographie, télévision), et la réalisation de prélèvements par biopsie de la muqueuse bronchique, de la muqueuse nasale, pharyngienne ou de la muqueuse du tube digestif.[WIK]*

L'endoscopie peut être utilisée, soit pour le diagnostic, soit pour traiter une maladie (endoscopie opératoire). Il s'agit d'un terme générique qui recouvre des examens spécifiques par organe. Ainsi, la bronchoscopie est l'exploration des bronches, la coloscopie, celle du côlon, la cystoscopie, celle de la vessie , etc.

### 1.3.3 Apport de l'informatique à l'endoscopie

Nous allons voir à présent l'apport que peut fournir l'informatique à l'endoscopie en terme d'imagerie.

#### 1.3.3.1 La coloscopie virtuelle (voir références [COLV] et [WIK])

La coloscopie virtuelle (coloscopie par tomodensitométrie) appelée aussi coloscanner est une méthode de dépistage avancée et hautement précise utilisée pour détecter le cancer du côlon. Sécuritaire, plus rapide et moins effractive que la coloscopie traditionnelle, cette technique consiste en un balayage de 60 secondes à l'aide d'une tomodensitométrie de l'abdomen suivi d'une reconstruction des images en trois dimensions qui permet de visualiser l'intérieur du côlon.

Une équipe de radiologistes analyse ces images pour déceler la présence de polypes dans le côlon. Les polypes sont de petites excroissances qui peuvent devenir cancéreuses si elles ne sont pas enlevées.

Pour plus de compréhension, voici une définition de la tomodensitométrie:

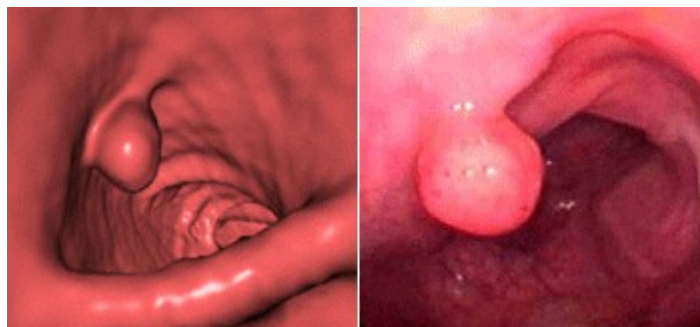
*La tomodensitométrie, tomographie axiale calculée , ou CT-scan est une technique d'imagerie médicale inventée officiellement par Godfrey Newbold Hounsfield : le patient est soumis au balayage d'un faisceau de rayons X. L'émetteur tourne autour du patient en même temps que les récepteurs chargés de mesurer l'intensité des rayons X après leur passage dans le corps.*

*Les données obtenues sont traitées par ordinateur, ce qui permet de reconstituer des vues en coupe des organes ou des vues en trois dimensions. On peut faire ressortir certains tissus, en particulier, les vaisseaux sanguins, en injectant un produit dit « de contraste » (souvent un complexe de l'iode), c'est-à-dire absorbant bien les rayons X et rendant très visibles les structures vasculaires (qui apparaissent hyperdenses).[WIKI]*

Voici une série d'image qui illustre la reconstruction d'un colon virtuel en 3 dimension à l'aide d'un logiciel et de l'information capturée lors du balayage des rayons X:



*Illustration 2: reconstitution 3D, à droite l'image de départ*



*Illustration 3: reconstitution 3D num 2, à droite l'image originale*

Veillez noter que la coloscopie virtuelle est une méthode de dépistage seulement. Comme c'est un examen non effractif, il est impossible d'effectuer des prélèvements de tissu ni d'enlever des polypes durant cette intervention. Si l'on découvre la présence de polypes, le patient devra subir une coloscopie traditionnelle.

Autre remarque, cette technique est actuellement encore en phase de développement, son efficacité varie largement en fonction de la taille des polypes, du logiciel et du matériel utilisé.

Un des gros avantages de la coloscopie virtuelle est qu'elle s'effectue, contrairement à la coloscopie traditionnelle, sans anesthésie générale. De plus, cela s'effectue plus rapidement qu'une endoscopie traditionnelle.

### 1.3.3.2 Endoscopie assistée par ordinateur

Il faut comprendre par endoscopie assistée, les techniques de traitements d'images qui permettent, lors de l'étude d'un patient, d'afficher en temps réel des informations complémentaires sur l'écran de façon à aider le praticien dans sa prise de décision. Ces informations concernent surtout l'ajout d'effets spéciaux afin, par exemple, de mettre en évidence des lésions qui seraient difficilement perceptibles ou distinguables en temps normal ou encore, la détection automatique de formes telles que des polypes ou autres anomalies.

Ces effets spéciaux peuvent prendre plusieurs natures, ils peuvent jouer sur la couleur, mettre en évidence le contour de certaines formes, attirer l'œil du scientifique en soulignant certains éléments de l'image ou encore, proposer une vue différente et complémentaire de ce qui est affiché dans la vidéo originale.

Voici deux exemples d'applications informatiques pouvant servir lors d'une endoscopie.

#### 1) Système d'aide pour une opération neuro-chirurgicale (voir référence [IGE])

Ici, l'application calcule à partir d'une image de départ (en haut à droite) une image en trois dimensions (en haut à gauche) qui met en évidence, en coloriant les zones concernées, la partie que va devoir opérer un neuro-chirurgien. Le programme propose aussi, en plus de ces deux images, d'afficher en même temps trois autres images donnant une vue tri-dimensionnelle de l'endroit où se trouve l'appareil d'endoscopie et sa direction.

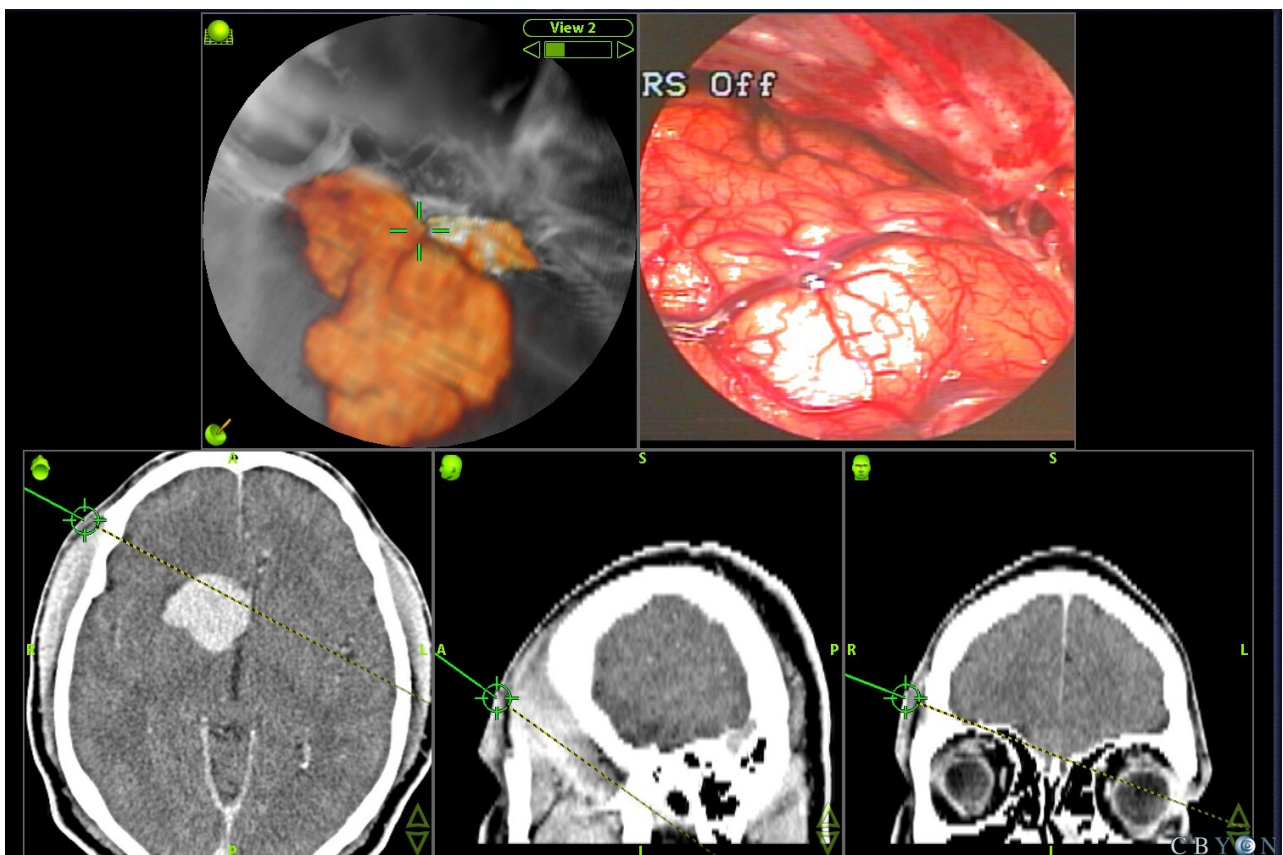
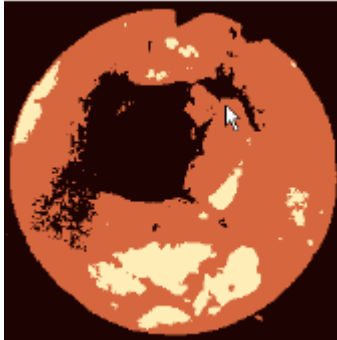


Illustration 4: Système d'aide pour opération

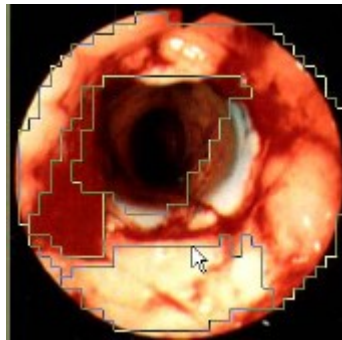
## 2) Système de reconnaissance de forme (voir référence [PMIA])

Dans cet exemple, une équipe de chercheur a développé un programme de calcul d'image qui permet d'extraire certains éléments de l'image originale. L'algorithme de calcul doit s'effectuer en parallèle pour obtenir un résultat dans un temps intéressant. La reconnaissance de forme s'effectue en trois étapes, présentées dans les images suivantes. Ces trois phases sont la segmentation de couleur, la segmentation des textures puis l'extraction des frontières du lumen(cavité d'un organe interne).

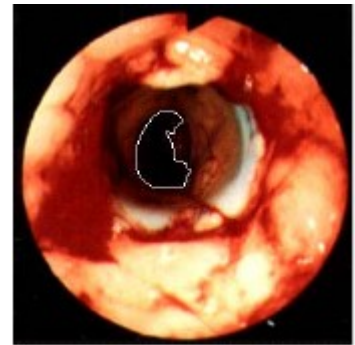
L'objectif ici est la détection du lumen central du colon.



*Illustration 5: Segmentation couleur*



*Illustration 7: Segmentation des textures*



*Illustration 6: Mise en évidence du lumen*

### 1.4 Inter-opérabilité, la norme DICOM

#### 1.4.1 Définition

Cette pluralité de sources d'images qu'engendrent les différentes disciplines médicales accroît la difficulté d'échanger des informations non seulement entre les différents départements interne à un hôpital mais aussi, entre différents hôpitaux. Si une personne vient à avoir un accident dans un lieu géographiquement éloigné de son hôpital habituel, il est important que ses antécédents cliniques ,et les informations qui y sont associées , puissent être communiqués le plus rapidement possible sur le lieu de sa prise en charge. Une autre difficulté est la transmission de nouvelles connaissances entre les différents centres de recherche, chercheurs et experts. Pour pouvoir s 'échanger des résultats, il faut d'abord se mettre d'accord sur la structure de ceux-ci et sur le protocole d'échange de données. C'est pour répondre à cette problématique d'inter-opérabilité entres différents systèmes d'informations que le standard DICOM à vu le jour.

DICOM (Digital Imaging and COmmunications in Medecine) est un standard de communication et d'archivage en imagerie médicale. C'est aussi par extension le format de fichier faisant référence dans le domaine de l'imagerie médicale.

Il a été créé en 1985 par l'ACR (American College of Radiology) et la NEMA (National Electric Manufacturers Association) dans le but de standardiser les données transmises entre les différents appareils de radiologie. Ce standard définit un format de fichier mais aussi un protocole de transmission des données (basé sur TCP/IP).

Pour plus d'informations sur DICOM, consulter la référence [DICOM]

Le réseau interne de l'Hôpital de Mont-Godinne supporte ce standard et une grande partie des outils et applications d'imageries médicales inter-agissent en DICOM.

#### **1.4.2 L'application Galerie**

Depuis l'année passée, le service de recherche informatique a démarré un nouveau projet, il s'agit de l'application Galerie, outil permettant de naviguer à travers les dossiers contenus sur le disque dur et, affichant dans une petite galerie une visualisation miniature des images JPEG contenues dans ce dossier. Elle permet en plus la sélection d'images et l'envoi de celle-ci à travers le réseau DICOM de l'hôpital.

### **1.5 La nécessité d'une application d'endoscopie**

Le service d'endoscopie de l'hôpital de Mont-Godinne ne dispose pas actuellement d'une application qui est intégrée directement dans le réseau interne. De plus, le logiciel actuel est un logiciel 'classique' fourni avec l'équipement d'acquisition vidéo installée sur l'ordinateur. En plus de poser des problèmes de droits de propriété, cette application est limitée car elle ne peut être modifiée afin de pouvoir interagir avec les autres applications d'imagerie ou le système d'information interne de l'hôpital. C'est pour ces raisons que le service a émis le souhait de disposer d'une application d'endoscopie qui serait développée au service de recherche informatique. Cette demande va mener cette année au lancement du projet nommé l'Endoscope Virtuel.

### **1.6 La nécessité d'une application d'intégration**

Nous avons vu qu'un hôpital est, à travers les services de médecines qui le composent, producteur d'une multitude d'images. Que ce soit le laboratoire de cytologie avec ses méga-images ou encore le service d'endoscopie avec ses vidéos, tous ces fichiers doivent être archivés afin de garder trace des diagnostics effectués pour un usage ultérieur lors par exemple du suivi d'un patient et de son historique de santé. Le problème actuel est que certaines de ces applications ne sont pas reliées au réseau interne DICOM, ce qui engendre une complication au niveau utilisateur car c'est à eux de prendre en charge la complexité engendrée par le manque d'intégration des logiciels. Un autre problème se situe au niveau de la dispersion des fonctionnalités. En effet, les applications étant jusqu'à lors développées séparément, chaque programmeur doit implémenter l'ensemble des fonctionnalités lui-même. Il serait plus intéressant de regrouper les fonctionnalités utiles à chaque application d'imagerie et de les rendre accessibles à tout nouveau projet, cela ferait gagner un temps précieux à l'équipe de développement. Par exemple, la fonctionnalité d'envoi DICOM des images JPEG de l'application Galerie serait très utile pour le Microscope Virtuel.

C'est afin de répondre à ces difficultés qu'un second projet d'interface commune est lancé cette année 2006-2007.

### **1.7 Conclusion**

Le service de recherche informatique de Mont-Godinne, au service des différents départements d'imagerie médicale, développe depuis plusieurs années, des applications informatiques afin d'optimiser et de faciliter le travail des scientifiques. Ces applications sont en partie développées par des mémorants dans le cadre de leur travail de fin d'étude, c'est le cas notamment du

Microscope Virtuel et de la Galerie. Cette année, afin de répondre à la demande du service d'endoscopie, le projet Endoscope Virtuel à vu le jour, cette application devra permettre aux praticiens de l'endoscopie de gérer plus facilement les fichiers capturés à l'aide de l'endoscope mais aussi, d'accéder à de nouvelles fonctionnalités.

Une autre difficulté est la dispersion des différents outils informatiques d'imageries, cela pose un problème au niveau de l'inter-opérabilité des différents programmes mais aussi, au niveau de la qualité des programmes et de leur temps de développement. Dans le but de résoudre ce problème, un second projet à vu le jour cette année, il s'agit du développement d'une plateforme d'acquisition multimodale.

La partie Objectif et Choix de ce mémoire fixe d'une façon plus concrète les objectifs à atteindre et expose les choix effectués qui ont permis l'implémentation de ces deux projets.

La partie Résultat présentera entre-autre la concrétisation de ces deux projets.

# Partie II Objectifs et Choix

## Introduction

Nous avons vu dans 'Etat de l'art' , la nécessité pour l' hôpital de Mont-Godinne de développer de nouveaux logiciels, ceux-ci lui permettront de rester à jour technologiquement par rapport à ce qui se fait dans le milieu médical.

Ce chapitre présentera les objectifs fixés pour l' année 2006-2007; ces objectifs ont principalement été posés par Monsieur Hubert Meurisse , maître de stage qui est également en contact permanent avec le corps médical, il est donc l' interprète entre le monde médical et le monde informatique. Pour mener à bien ces projets d' Endoscope Virtuel et d' intégrations des outils d' imagerie, des choix ont été entrepris , ils seront exposés à la suite des objectifs de chaque applications; ces choix concernent aussi bien les technologies que les principes à suivre pour la modélisation ou encore le matériel de test utilisé .

## Chapitre 2 Une plate-forme d'intégration

### 2.1 Présentation du projet

L' objectif ici, est de créer une plateforme qui intègre toutes les applications d' imagerie existantes( le Microscope Virtuel et la Galerie d' images), les applications en cours de développement(Endoscope Virtuel) et, permet aussi aux applications futures de s' intégrer facilement.

Nous donnerons à ce projet de plateforme le nom d' Interface Commune. Le mot interface prend ici deux sens. Premièrement, au niveau utilisateur, un sens 'graphique', c' est à dire que l' utilisateur pourra dans une seule application accéder aux fonctionnalités de tous les outils d' imagerie.

Deuxièmement au niveau du programmeur, cela signifie une application capable d' intégrer les différents composants existant entres-eux, de façon à ce qu'ils puissent échanger de l' information , être accédé en même temps et utiliser des fonctionnalités communes.

Les objectifs pour ce projet étant d' assez haut niveau, ils sont exposés plus en détail dans les trois points suivant.

### 2.2 Une interface ergonomique et intuitive

L' application étant conçue de façon générale pour servir au monde médical,cela implique une interface pensée de façon ergonomique et intuitive. En effet, l' utilisateur doit pouvoir naviguer facilement et simplement entre les différents outils qui lui sont proposés. Par exemple, le passage du Microscope Virtuel à l' Endoscope Virtuel doit être simple et rapide. De plus, on doit pouvoir travailler sur un seul outil à la fois. Par exemple, si le Microscope virtuel et la Galerie d' image sont présent, il doit être possible de travailler uniquement sur la Galerie pendant que le Microscope Virtuel est présent mais ne fonctionne pas(n' utilise pas de capacité de calcul de la machine). D' un autre côté, l' utilisateur doit pouvoir faire fonctionner le Microscope Virtuel et, pendant le traitement d' image, travailler sur la Galerie.

A ces fins, l'interface graphique devra comprendre un système d'onglets, chaque onglet comportera un outil d' imagerie, l'utilisateur pourra ainsi passer d'un outil à l'autre de façon aisée et rapide. Le dernier onglet servira pour les fonctionnalités DICOM, ce système demande une architecture dans



laquelle chaque outil est indépendant des autres.

Cela nécessite une conception qui soit la plus décomposable possible afin de pouvoir tester différentes configurations graphiques et de modifier facilement l'application.

### **2.3 Ajouter et rassembler des fonctionnalités communes**

L'application devra permettre à tout les programmes intégrés d'accéder à des fonctionnalités communes. Ces fonctionnalités sont pour l'instant l'acquisition et l'envoi d'informations sur le réseau DICOM.

Il faut bien entendu que les différentes applications soient cohérentes entre elles. Par exemple, elles travaillent sur le même patient ou, envoient leurs fichiers acquis sur le même compte 'patient' dans le réseau DICOM de l'hôpital.

Les différentes applications doivent pouvoir aussi accéder aux fonctionnalités propres à chaque type de fichier. Par exemple, l'application Galerie permet actuellement de visualiser les fichiers jpeg dans des petites icônes JAVA; toutes les applications doivent pouvoir faire de même sans pour autant réécrire plusieurs fois le même code dans chaque application.

### **2.4 Une conception qui permet flexibilité et maintenabilité**

Ce projet étant à son point de départ, il est fort probable que de nouvelles fonctionnalités et de nouveaux outils devront, dans un futur plus ou moins proche, être ajoutés à l'Interface Commune.

La conception doit dès lors permettre une grande flexibilité et maintenabilité<sup>1</sup>.

Flexibilité dans le sens où il devrait être facile d'enlever ou de modifier certains composants actuels voir de présenter plusieurs interfaces graphiques différentes de façon à pouvoir permettre un test de l'ergonomie par un utilisateur potentiel.

La maintenabilité signifie la possibilité, sans modifier de façon conséquente le code actuel, d'ajouter de nouveaux outils ou de nouvelles fonctionnalités.

## **Chapitre 3 Choix effectués pour l'Interface Commune**

### **3.1 Re-conception et décomposition de l'application 'Galerie'**

En ce qui concerne les fonctionnalités de demande et d'envoi d'information sur le réseau DICOM, l'application actuelle 'Galerie', présentée brièvement dans la partie 'Existant de l'hôpital', implémente déjà ces deux fonctionnalités. Il serait donc dommage de ne pas sans servir.

Cependant, la conception de ce programme, bien qu'il soit écrit en JAVA ne suit pas les principes de l'orienté-objet. Il faut donc d'abord passer par une phase d'étude du logiciel si l'on veut non seulement récupérer les fonctionnalités DICOM et les ajouter dans l'interface mais aussi y ajouter la Galerie elle-même. Il s'agit donc dans un premier temps de déterminer le diagramme de classe car il n'y a pas de documentation fournie actuellement, puis, dans un deuxième temps, il faut re-conceptualiser le programme pour en détacher les fonctionnalités DICOM et les insérer dans l'Interface en tant que fonctionnalités communes.



## **Chapitre 4 Un Endoscope Virtuel**

### **4.1 Présentation du projet**

L'objectif à long terme est un endoscope virtuel écrit en JAVA capable de prendre en entrée les informations envoyées par l'appareil d'endoscopie de l'hôpital et, de fournir en output la vidéo sur l'écran de l'utilisateur. A cela s'ajoute aussi différentes fonctionnalités telles que l'enregistrement dans un format numérique, la communication avec le réseau DICOM ou encore le traitement d'image (l'ajout d'effet visuel sur la vidéo pour aider l'utilisateur à interpréter ce qu'il voit à l'écran). Ce projet sera nommé Endoscope Virtuel

### **4.2 Objectifs 2006-2007**

Le projet étant à son point de commencement et, le traitement vidéo avec le langage JAVA n'étant pas encore d'usage courant, le principal objectif pour cette année 2006-2007 sera de fournir un prototype qui permet de comprendre les limites actuelles et les possibilités que l'on peut réaliser en matière de vidéo avec JAVA.

Nous allons essayer dans un premier temps de simplement arriver à acquérir de l'image vidéo provenant d'un input (fichier ou appareil) et, à partir de cette information, pouvoir enregistrer un fichier vidéo et des snapshots au format jpeg.

Il faut bien entendu une certaine liberté habituelle à tout lecteur vidéo au niveau de l'interface: lecture, pause, avancer, reculer, arrêter, changer d'appareil,...

Cet outil doit bien entendu être intégré dans l'Interface commune et, doit pouvoir communiquer avec le réseau DICOM. Cela implique le codage des vidéos capturées au format DICOM.

### **4.3 Une modélisation qui permet maintenabilité et simplicité**

En ce qui concerne la modélisation, les critères retenus sont la simplicité et la maintenabilité. On ne doit pas connaître en détail le package JMF pour pouvoir comprendre le code. De plus, le projet étant à son point de démarrage, la conception doit permettre aux programmeurs suivants de pouvoir ajouter de nouvelles fonctionnalités sans pour autant changer le code; par exemple l'enregistrement dans un autre type que le mpeg2 ou encore l'ajout d'effets visuels sur la vidéo. Bien sûr la modélisation doit aussi permettre à l'application de s'intégrer parfaitement dans l'Interface Commune.

## **Chapitre 5 Choix effectués pour l' Endoscope Virtuel**

### **5.1 Le matériel de test**

Les standards vidéos étant nombreux et, les différents systèmes d'exploitations proposant chacun leur propre format ainsi que leur propre driver; il était important de pouvoir recréer en laboratoire des conditions qui soient le plus proche possible de la réalité. D'un autre côté, le matériel disponible étant limité et, le projet n'étant pas encore assez avancé pour travailler directement sur l'appareil d'endoscopie de l'hôpital, nous avons opté pour le matériel de test suivant:

Un système Pinnacle prenant en entrée une source vidéo analogique et fournissant en sortie un signal digital, le boîtier Pinnacle est raccordé à l'ordinateur par une carte FireWire. La source vidéo étant un magnétoscope.

Un second matériel de test est une webcam qui se branche via le port usb de la machine; l'ordinateur de test est imposé, et il utilise l'os windows 2000. Nous pouvons donc à priori prendre en compte tout type de matériel puisque nous testons aussi bien l'entrée usb que l'entrée FireWire. Les fiches techniques du système Pinnacle et de la webcam sont présentées dans l'Annexe I

## **5.2 L' API JMF**

En ce qui concerne le software, JAVA étant imposé, c'est le package 'Java Multimedia Framework' qui a été choisi parmi d'autres car il présente plusieurs avantages.

Premièrement, il a été conçu par Sun Microsystems ce qui garantit une compatibilité optimale avec l'environnement JAVA. Deuxièmement, il n'y a pas besoin de rajouter manuellement ou via la compilation de code autre que JAVA (par exemple c, c++) des drivers ou des fichiers système, tout se fait automatiquement lors de l'installation (à l'inverse des autres packages trouvés sur le net). Troisièmement, il peut à priori lire tout type d'entrée vidéo qui utiliserait le standard windows VFW, cela signifie que l'on peut un théoriquement travailler avec tout appareil numérique qui serait reconnu par windows. De plus, il permet à l'heure actuelle le plus de fonctionnalités mais aussi la lecture d'un grand nombre de type de fichiers vidéos.

## **5.3 L'enregistrement en MPEG2 MP@ML**

Parmi la multitude des formats possibles pour l'enregistrement vidéo, c'est le MPEG2 MP@ML qui a été préféré et cela, pour la raison simple qu'il est la norme vidéo utilisée dans le standard DICOM (voir [DICOM]). De plus, le mpeg2 est aussi la norme utilisée dans la télévision internet mais aussi dans l'industrie du DVD. Il présente donc un avantage certain pour le futur de l'application en ce qu'il a une grande polyvalence. Pour plus d'explications, une étude sur ce format est proposée dans la partie Résultat.

# **Chapitre 6 Introduction à JMF**

Ce chapitre est inspiré de la référence [JMF]

## **6.1 Introduction**

Java Media Framework (JMF) est une API large et transparente utilisée pour les applications multimédias (Son / Vidéo). La version actuelle de JMF est la 2.1. Cette API est une initiative de SUN qui souhaite apporter une solution «time-based media processing» (processeur de media basé sur un timeline) à Java. Les média basé sur le temps sont des données qui changent par rapport au temps. Nous les retrouvons bien entendu dans les vidéos et l'audio.

## **6.2 Possibilités**

On peut faire un ensemble impressionnant d'applications autour du multimédia avec JMF. Voici un ensemble d'idées d'applications pouvant être réalisées en Java :

- Lecture de différents formats multimédia dans une applet Java. Les formats supportés sont AU, AVI, MIDI, MPEG, QUICKTIME et WAV.
- Lecture d'un flux de données (média) depuis internet.
- Capturer le son et la vidéo depuis un microphone et/ou une caméra et enregistrer les données dans un format supporté.
- Modifier les flux audio/vidéo en temps réel.

- Transmettre de l'audio ou de la vidéo en temps réel sur internet.
- Broadcaster de la musique / vidéo comme les radio ou les programmes télévisés.

Cependant, avant de pouvoir utiliser JMF, on doit comprendre son architecture et les concept clés qui composent ces principales classes.

### 6.3 Architecture

Nous pouvons prendre le système stéréo comme comparaison à JMF. Lorsque l'on écoute un CD de musique avec un lecteur de CD, c'est le CD qui fournit les données au système.

Cependant ces données ont été, auparavant, capturées par des microphones et autres périphériques dans les studios d'enregistrement. Le microphone est donc utilisé comme périphérique de capture alors que le CD est simplement la source de données.

Le lecteur de CD lit et joue la musique grâce aux enceintes (périphérique de sortie). Cependant, vous pouvez tout à fait connecter des écouteurs à votre lecteur de CD. Dans ce cas là, ce sont les écouteurs qui représentent le périphérique de sortie.

JMF utilise le même modèle objet, nous retrouvons donc:

- Data source
- Capture device
- Player
- Processor
- Datasink
- Format
- Manager

Nous allons voir en détail chacun de ces objets.

#### 6.3.1 Data Source

La source de données (DataSource) encapsule le flux du média comme le CD de musique. Avec JMF, une DataSource représente le média audio, vidéo ou une combinaison des deux. Cette DataSource peut aussi bien être un fichier ou venir d'un flux internet. L'avantage principal de cette classe est qu'une fois la provenance et le protocole définis, l'objet encapsule automatiquement la provenance et le protocole utilisé pour délivrer le média. Une fois créée, la DataSource peut être couplée à un Player afin d'être jouée ! De ce fait, celui-ci n'a pas à se soucier d'où vient la source et de sa forme initiale...

Vous pouvez récupérer les données de une variété de sources comme depuis un fichier local ou réseau ou encore «un broadcast internet en live».

Vous devrez classer la datasource utilisée suivant le type d'initialisation du transfert de données :

- Pull Data Source : le client initialise le transfert de données et contrôle le flux de données depuis la source. (HTTP et Fichier serveur sont de bons exemples pour ce type d'initialisation).
- Push Data Source : le serveur initialise le transfert de données et contrôle le flux de données depuis une source spécifique (broadcasts principalement).

Bien entendu, nous verrons que l'on peut combiner les Data sources en une seule. Si vous souhaitez capturer une scène en live, vous devez avoir deux sources : audio et vidéo. Dans cette situation, vous voudriez combiner ces deux sources en une seule pour faciliter le contrôle.

#### 6.3.2 Capture Device

Cet objet représente un périphérique matériel que vous pouvez utiliser comme source de données. Cela peut être un microphone, un appareil photo ou une caméra. La source de données peut être

connectée à un Player permettant ainsi d'être affichée à l'écran ou d'être traitée afin d'être convertie dans un format spécifique ou encore sauvegardée pour une utilisation ultérieure. Ces périphériques peuvent être également classés en «push» ou «pull» périphériques. Avec une source de type «pull», l'utilisateur contrôle quand il souhaite effectuer la capture (par exemple un appareil photo). A l'inverse, un microphone est de type «push» car il fournit en continu un flux audio.

### 6.3.3 Player

L'objet Player utilise un flux audio ou vidéo en entrée et le rend (l'affiche sur l'écran pour une vidéo et le joue sur les enceintes pour le son). De la même façon qu'un lecteur DVD lit les données et les affiche sur la télévision et joue le son sur les haut parleurs.

Un «Player» est un objet avec état. En effet, il doit se préparer ainsi que la source de données avant même de pouvoir commencer à la lire. Si vous insérez un DVD dans votre lecteur, celui-ci ne vous lit pas instantanément le film, mais il commence par chercher la première piste et effectue d'autres initialisations... Finalement, après quelques secondes, vous pouvez voir le film !

Voici l'ordre des états par lequel passe le «Player» :

- Unrealized : dans cet état, le «Player» est instancié. A ce moment, celui-ci ne connaît rien du média à lire ...
- Realizing : le Player passe dans cet état lorsque vous appelez sa méthode *realize()*. Le player traite l'ensemble des ressources nécessaires.
- Realized : dans cet état, le Player sait précisément les ressources dont il a besoin et a les informations concernant le type de média qu'il va utiliser. Il peut également fournir un ensemble de composants visuels et contrôle. L'ensemble des connexions avec les composants systèmes est établi.
- Prefetching : lorsque l'on appelle la méthode *prefetch()*, le Player se positionne de l'état Realized vers Prefetching. Cet état déclare que le média est prêt à présenter le média. Pendant cette période, le Player pré-charge les données de son média, obtient les ressources exclusives et fait l'ensemble des traitements utiles à la lecture du média.
- Prefetched : l'état dans lequel le Player a terminé le Prefetching. Le média est prêt à démarrer.
- Started : le Player entre dans cet état lorsque vous appelez sa méthode *start()*, le média est alors prêt à présenter les données du média.

### 6.3.4 Processor

Un Processor est un type particuliers de Player (l'interface Processor hérite de Player). Vous pourrez, avec un Processor, contrôler et modifier les données en entrée et/ou en sortie.

En plus des états indiqués pour le Player, le Processor en intègre deux autres :

- Configuring : un Processor entre dans cet état à partir de l'état unrealized et lorsque la méthode *configure()* est appelée. Un Processor est également dans cet état lorsqu'il se connecte à la DataSource, de multiplexe le flux d'entrée, et accède aux informations liées au format des données d'entrée.
- Configured : un Processor entre dans cet état lorsqu'il est connecté à la DataSource et lorsque le format est déterminé.

### 6.3.5 DataSink

L'objet DataSink est l'interface de base pour lire un média à partir d'une DataSource ou envoyer le résultat vers une destination. DataSink peut être utilisé pour écrire un média dans un fichier.

### 6.3.6 Format

Un objet Format représente un format de média précis. Il décrit le nom de l'encodage et le type de données que le format accepte. Il existe deux classes principales héritant de Format:

- AudioFormat: format audio
- VideoFormat: format video

### 6.3.7 Manager

Le Manager est un objet intermédiaire que l'on ne peut réellement comparer à la réalité. C'est une classe qui vous permettra de connecter deux différents objets. Vous pourrez, par exemple, créer un Player à partir d'une DataSource.

JMF fournit 4 types de Manager:

- Manager : à utiliser pour créer des Player, Processor, DataSource, DataSink.
- PackageManager : ce manager gère un registre de package contenant les classes JMF, telles que les Player, Processor, DataSource, DataSink personnalisés.
- CaptureDeviceManager : ce manager contient le registre des périphériques disponibles

PlugInManager : ce manager maintient un registre des plug-in JMF (composants de traitement) disponibles

## 6.4 Installation et enregistrement d'un périphérique

JMF n'étant pas simplement une suite de bibliothèques JAVA, mais un outil à part entière, il faut, pour permettre la communication entre l'appareil d'entrée et l'application, l'installer puis le configurer correctement .

### 6.4.1 Installation

Avant tout, télécharger le framework à l'adresse suivante: <http://java.sun.com/products/java-media/JMF/index.jsp>. Ensuite, lancer l'exécutable et suivre les instructions du programme qui installera sur l'ordinateur, non seulement les bibliothèques JAVA (fichiers JAR) mais aussi, un outil appelé 'JMFRegistry' permettant entre autres, l'enregistrement d'un périphérique et, la connection de celui-ci avec les classes JMF.

### 6.4.2 Enregistrement d'un appareil de capture

Afin de pouvoir utiliser un appareil extérieur à l'ordinateur dans un programme JMF, il faut que celui-ci soit reconnu par windows mais aussi par JMF. Cela est rendu possible en enregistrant l'appareil dans l'utilitaire 'JMFRegistry'.

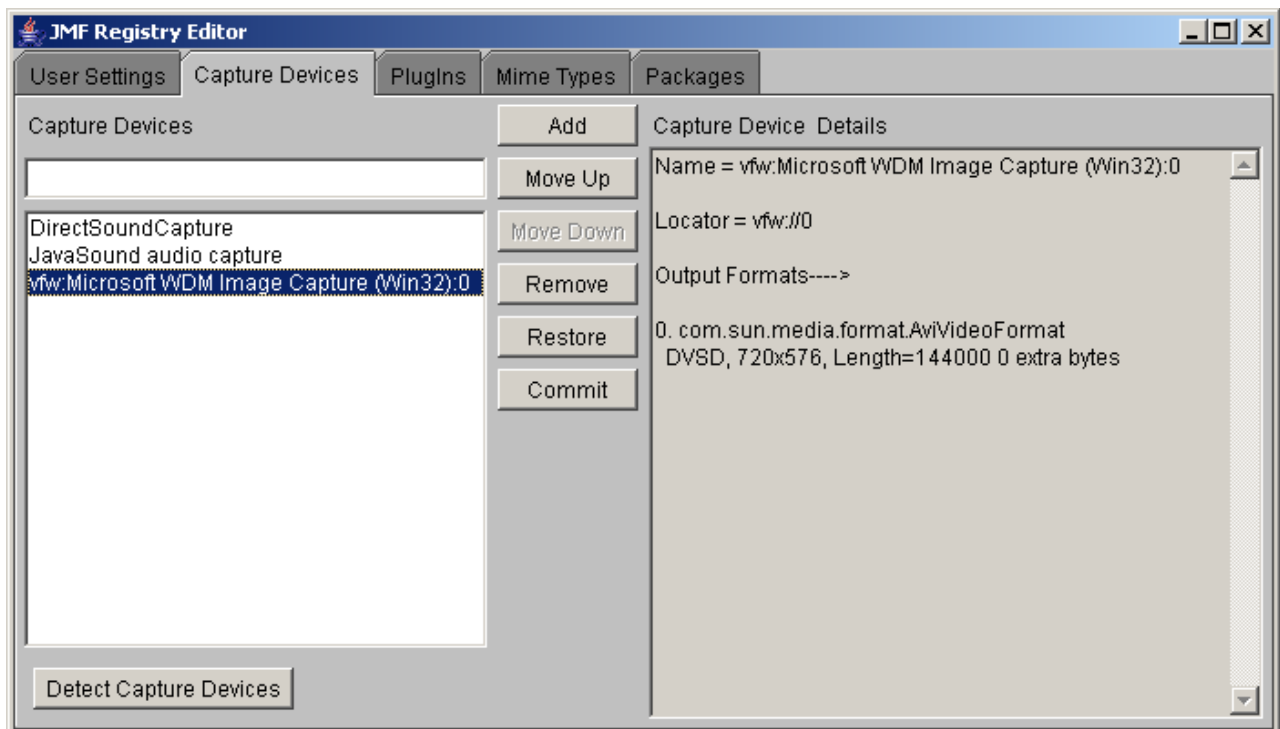
Dans un premier temps, l'appareil doit être connecté à l'ordinateur et reconnu par windows. C'est à dire qu'il doit apparaître dans les informations windows concernant les périphériques. Il faut s'assurer ensuite qu'aucun programme ou processus windows n'utilise cette ressource périphérique sinon, JMF ne pourra pas le détecter (cela vaut aussi lors de l'utilisation de l'appareil avec un programme tournant avec JMF).

Une fois le périphérique connecté, on doit lancer JMFRegistry, dans l'onglet 'capture device', cliquer sur le bouton 'detect capture devices'. L'utilitaire affichera alors un nouvel outil dans la

fenêtre de gauche. Il ne reste plus qu'à cliquer sur le bouton 'Commit' pour enregistrer l'outil de capture dans le JMFRegistry.

On peut maintenant faire appel à l'appareil dans tout programme utilisant les librairies de JMF.

Voici un exemple qui présente un outil enregistré dans JMF:



*Illustration 8: JMFregistry*

Pour plus de détail, l'Annexe II propose la liste des périphériques vidéos reconnus par JMF.

# Partie III Résultats

## Introduction

Dans le chapitre précédent, nous avons vu quels étaient les objectifs à atteindre , nous avons pour cela effectué une série de choix appuyé d' une argumentation, cette partie exposera les résultats obtenu à l' aide de ces décisions.

Il sera présenté dans un premier temps les deux parties applicatives avec, d'abord l' interface commune puis l' Endoscope Virtuel; pour chacune de ces applications une description de la modélisation et des concepts quelle renferme sera présentée suivie, par une série de 'screenshots' qui illustrerons les fonctionnalités implémentées et l' ergonomie de l' interface. Les objectifs atteints ou non par leurs implémentation seront soulignés afin de permettre au lecteur une vision plus claire en rapport avec ce qui à été fixé précédemment.

Le chapitre se clôtura par une étude sur le format vidéo qu'est le MPEG2 MP@ML ,standard utilisé dans la norme DICOM mais aussi dans l' industrie de la télévision interactive et du dvd.

## Chapitre 7 L'interface commune

### 7.1 Introduction

Cette partie présente la réalisation du projet intitulé 'Interface commune' , c'est d'abord la conception en diagramme de classe qui est présentée , suivie par une description des classes exposées dans celle-ci. Par un souci de lisibilité et de compréhensibilité, deux diagrammes seront exposés. Le premier porte sur la modélisation de l'application d'un point de vue général tandis que le second, se concentre sur le concept de 'Source'.

Des 'snapshots' commentés sur leur contenu montreront ensuite d'une façon plus concrète le résultat final obtenu pour ce projet .Ce chapitre se conclura par des diagrammes de séquences qui expliqueront plus en détail l'interaction entre les différents composants du programme.

### 7.2 Diagramme de classe général

Le diagramme de classe est un schéma utilisé dans l'ingénierie du logiciel afin d'exprimer les différentes classes d'un système informatique ainsi que les relations entre elles. Il précède la phase d'implémentation et le programmeur se base sur cette conception pour coder le programme .Chaque classe représente un groupe d'objets qui joueront un rôle particulier au sein de l'application. C'est ce schéma qui est présenté dans la figure qui suit:

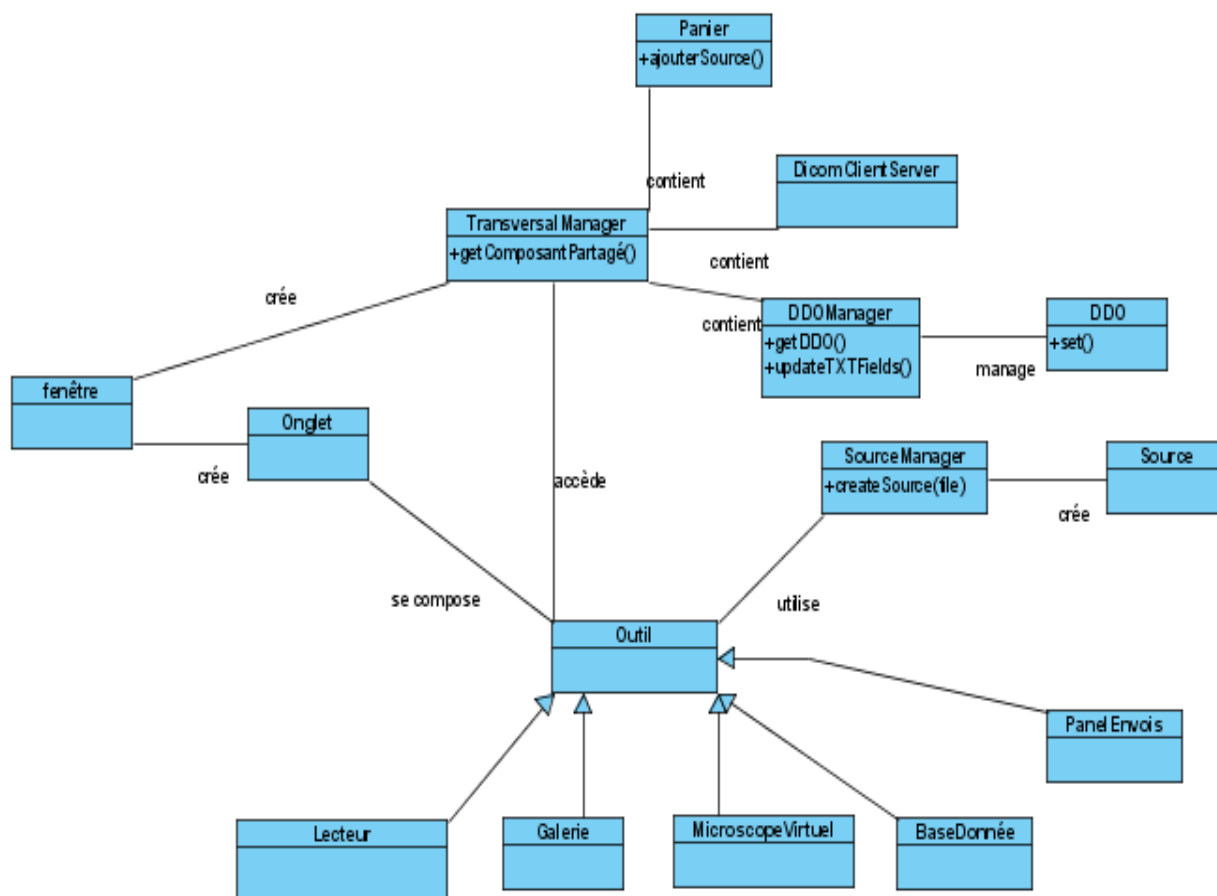


Illustration 9: Diagramme de classe général

### 7.3 Explication du diagramme

Le schéma ne suffisant pas à lui seul pour expliquer clairement le rôle joué par chaque classe, voici une explication sur le 'pourquoi' de la présence des classes, leur responsabilité, ainsi que les concepts dont elles proviennent.

#### 7.3.1 Les classes 'Fenêtre' et 'Onglet'

La classe Fenêtre représente simplement la fenêtre 'windows' dans laquelle tourne l'application. La classe 'Onglet' quant à elle est le système d'onglet dans lequel les outils d'imagerie sont affichés.

#### 7.3.2 Le concept de 'Outil'

Ce concept a été modélisé par la classe du même nom, Outil. Elle représente un outil d'imagerie à part entière. Par exemple, le Microscope Virtuel, le Lecteur vidéo ou encore le Panel Envois sont des enfants de cette classe. La classe étend l'objet JAVA 'Jpanel' et donc hérite des méthodes qu'elle contient. Cela signifie que l'on peut afficher les outils dans les différents onglets de l'interface graphique. Lorsqu'un programmeur voudra insérer un nouvel outil d'imagerie, c'est cette classe qu'il



devra implémenter .Voici une explication plus approfondie de ceux-ci:

#### La classe Galerie

Outil de navigation à travers le disque dur, il affiche sous forme d'une galerie d'images tout les fichiers d'un dossier sélectionné.

#### La classe Lecteur

Outil qui permet de sélectionner une source vidéo(fichier, appareil de capture)et, en fonction, fait appel à un lecteur particulier qui affichera le contenu à l'écran. C' est en fait la réalisation du projet de prototype d' Endoscope Virtuel décrit dans le chapitre suivant.

#### La classe PanelEnvois

Permet à l'utilisateur de visualiser les fichiers capturés par les différents outils d'imagerie. Il affiche aussi les informations relatives au contexte. Ces informations sont les données relatives au patient et les destinations possibles pour l'envoi de données DICOM. C'est cet outil qui permet de regrouper les fonctionnalités DICOM et les rend disponibles aux autres outils, comme souhaité dans les objectifs.

#### La classe MicroscopeVirtuel

Représente le Microscope Virtuel, outil présenté brièvement dans la partie Etat de l'art

#### La classe BaseDonnée

Outil développé cette année de stage 2006-2007 par Cédric Peeters et qui permet la récupération et l'affichage d'images stockées dans un fichiers de base de donnée.

### 7.3.3 Le concept de 'composant partagé'

Ce concept n'est pas modélisé par une classe en particulier mais regroupe un ensemble de classes, ce sont des composants qui fournissent aux outils d' imageries des fonctionnalités communes et nécessaires pour faire fonctionner l'application. Par exemple, l'accès au réseau DICOM, le stockage et la gestion des informations contextuelles ou encore , la gestion des images capturées. Ces composants répondent à l'objectif de communication entre les outils. Voici une explication des différents composants partagés:

#### Le TransversalManager

Il contient et gère les différents composants partagés(Panier, DicomClientServer, DDOManager), c' est lui qui les instancie . Chaque composants outil doit avoir en paramètre lors de sa création la même instance de ce TransversalManager, cela garantit l'unicité et l'accès pour les outils aux composants partagés .

#### SourceManager

Manager qui permet de créer une Source, il suffit de lui passer en paramètre un objet du type File, il se charge de créer la Source correspondante en fonction du type de l'objet File. La méthode permettant d'initialiser un objet Source est accessible par toutes classe, même si celle-ci ne possède pas de référence vers le SourceManager(on dit alors que la méthode est 'static').

## Source

Une Source est un objet qui contient un fichier d'imagerie produit par l'un des outils de l'application.

## Le Panier

Objet qui doit être accessible par tout les outils et composants partagés. Il ne peut y avoir qu'une seule instance de cette classe. Il gère un tableau dans lequel les outils déposent des objets appartenant à la classe 'Source' ou qui l'étendent.

Cette classe propose aussi un ensemble de méthodes pour gérer les 'Sources' quelle contient (ex: vider le panier,visualiser une source,...).

Les autres composants ajoutent des 'Sources' dans ce Panier en

- 1) créant les 'Source' via le SourceManager
- 2) se procurant l'instance du Panier
- 3) en appelant la méthode du Panier qui permet d'ajouter la ou les 'Source' et en passant la 'Source' en paramètre.

Ce composant étend la classe Jpanel, cela permet à l'outil PanelEnvois d' afficher le contenu du panier sous forme d' une galerie d' image et de permettre à l'utilisateur de visualiser le contenu de ce panier.

## Le DicomClientServer

Objet qui permet de communiquer avec le réseau DICOM, il est fournit dans la librairie 'DICOM.jar',créé par l'équipe de développement de l'hôpital.

## Le DicomDataObject ou DDO

Objet contenant au format DICOM des informations sur un patient ou un examen.

## Le DDOManager

Stocke les différents DicomDataObject (objets contenant au format DICOM les données relatives à un patient). Il étend aussi la classe Jpanel, afin de permettre à l'outil PanelEnvois d'afficher les données relatives au patient. Il propose aussi ,dans son affichage, des fonctionnalités qui permettent à l'utilisateur de gérer ces informations.

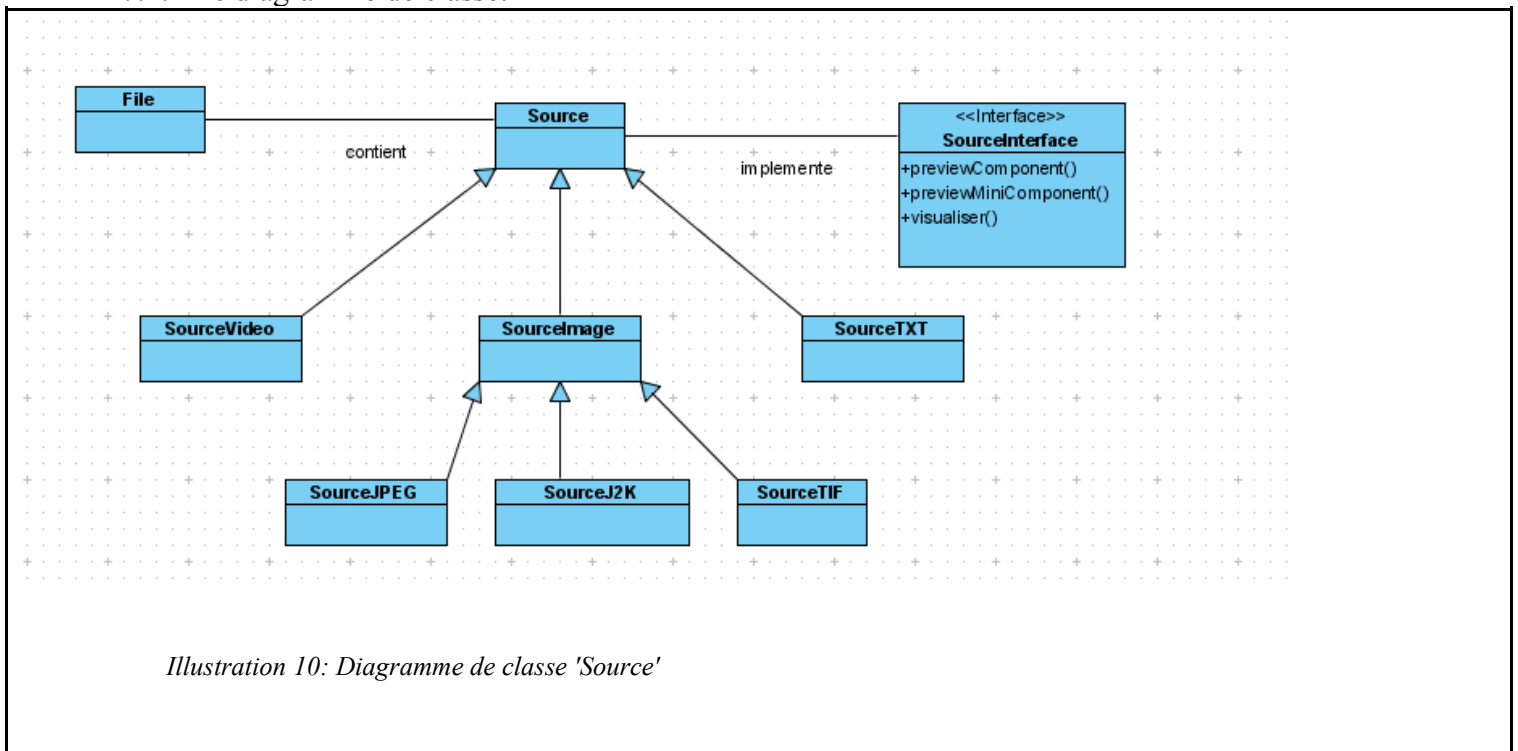
## 7.4 Le concept de 'Source'

### 7.4.1 Explication du concept

Dans le premier diagramme, il y a une classe nommée 'Source' qui n'a pas fait l'objet de commentaires approfondis. En effet, ce concept nécessite non seulement une explication plus vaste que les autres composants mais aussi, un diagramme de classe se focalisant uniquement sur cette classe et ses sous-classes.

Une Source est en fait un fichier de n'importe quel type, qui est actuellement utilisé par un des outils de l'application, plus une série de fonctionnalités de base qui sont accessibles par n'importe quel outil. Elle contient en attribut un objet JAVA du type 'File'(le fichier en question) et, possède comme méthodes les fonctionnalités propres à ce type de fichier. Évidemment, ces fonctionnalités de base ne peuvent être implémentées de la même façon pour tout les types de fichier (on affiche pas une image comme on affiche une vidéo,...), la classe parente 'Source' est étendue par des classes qui représentent un type de fichier en particulier(SourceVideo,...) et , qui implémentent les méthodes en fonction du fichier qu'elles représentent. Cette classe répond à l'objectif où, les outils d'imageries accèdent à des fonctionnalités propres à chaque type de fichier, sans devoir les implémenter eux-mêmes.

### 7.4.2 Le diagramme de classe:



### 7.4.3 Explications des classes:

Afin de comprendre ce qu'apportent les classes du schéma dans l'application, un commentaire pour chacune d'entre-elle s'impose: .

#### SourceInterface

Cette interface représente les fonctionnalités que doit fournir toute classe ou sous-classe de Source. Cela permet une certaine homogénéité; lorsqu' un morceau de code utilisera une Source, il saura qu'il peut compter sur ces fonctionnalités.

File

Représente un fichier stocké sur disque. Cette classe est comprise dans les librairies de base JAVA

Source

Représente un fichier auquel est associé une série de fonctionnalité. Voir le point 8.3

SourceVideo, SourceJPEG, SourceJ2K, SourceTIF, SourceTXT

Classes représentant un fichier d'un certain type, les méthodes qu'elles proposent sont implémentées de façon à correspondre au type particulier du fichier qu'elles contiennent. Par exemple, SourceJPEG contient une image au format JPEG et des méthodes propre à ce type, qui ne pourraient être utilisées pour un fichier vidéo, on ne visualise pas une image JPEG de la même façon qu'une vidéo.

## 7.5 Screenshots de l'application

Voici visuellement ce qu'à donné l'implémentation en JAVA de la modélisation exposée plus haut. Chaque image est suivie d'une ou plusieurs remarque(s) sur son contenu soit, par souci de clarté soit, pour faire le lien entre les éléments de l'interface graphique et les concepts exposés dans les diagrammes de classes.

### 7.5.1 Vue d'ensemble avec l'outil 'Galerie'

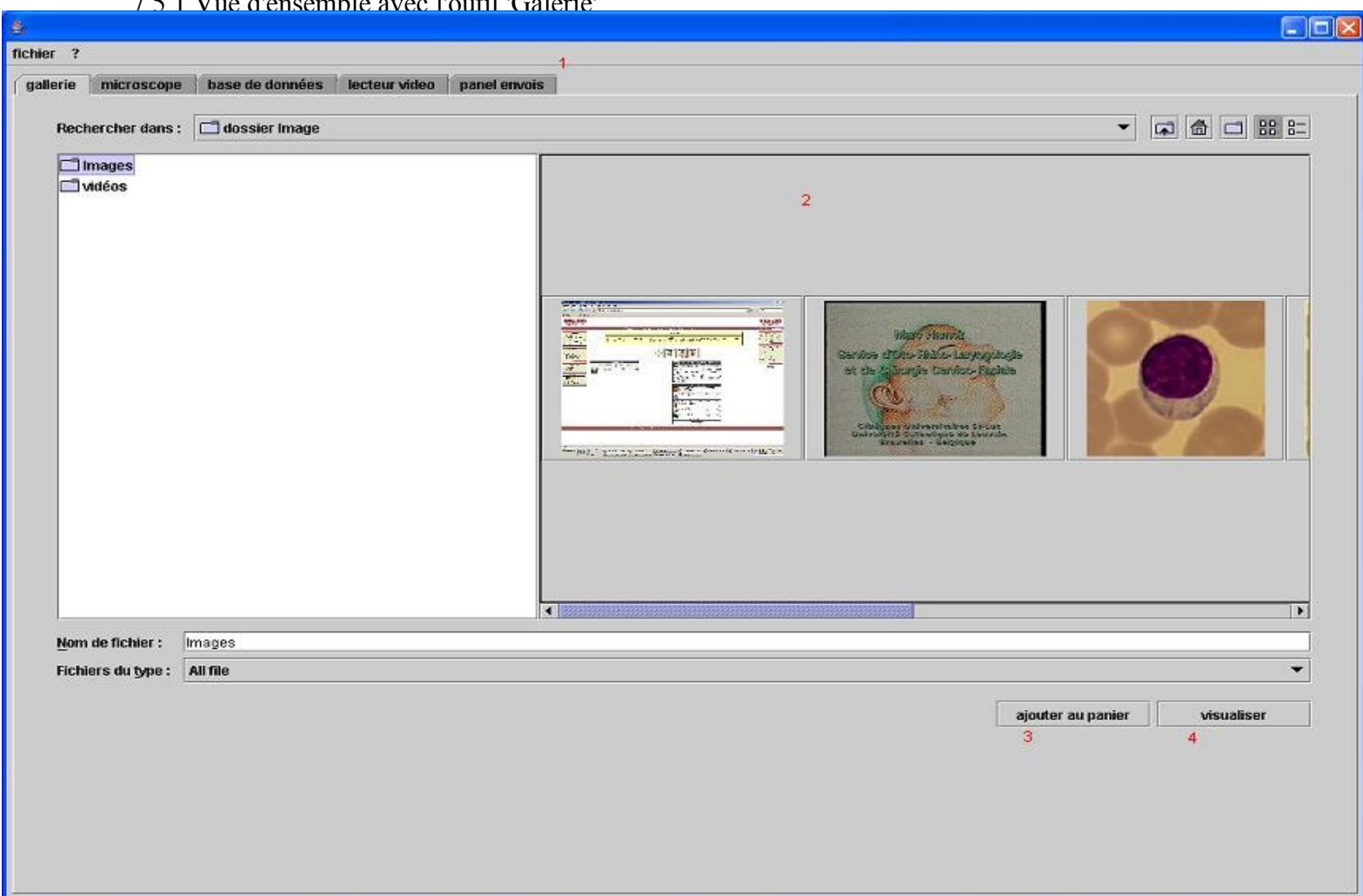


Illustration 11: L'outil Galerie

1) Les onglets: chaque onglet représente un outil d'imagerie, la Galerie, le Microscope, la Base de Données, le Lecteur vidéo et le Panel Envois DICOM. Pour passer d'un outil à l'autre, il suffit de cliquer sur l'onglet correspondant, cela ne stoppera pas les tâches de l'outil sur lequel vous travaillez.

2) La prévisualisation d'images: C'est une galerie d'images où chaque icône représente un fichier du dossier sélectionné dans le navigateur à gauche de l'écran. Cette icône est obtenue grâce à la méthode 'previewMiniComponent()' de la classe Source correspondante. Par exemple de la classe SourceJPEG pour un fichier JPEG.

3) Le bouton ajouter au panier: Lorsque l'on clique sur ce bouton, il envoie dans le panier les fichiers qui ont été sélectionnés dans la galerie. Le panier se trouve dans l'onglet Panel Envois, qui sera illustré dans le screenshot suivant.

4) Le bouton visualiser: Lorsque l'on clique sur ce bouton, une fenêtre s'ouvre et affiche le ou les fichiers qui ont été sélectionnés dans la galerie. Cette visualisation du fichier dépend évidemment du type de fichier par exemple, si c'est une image, elle sera affichée sous sa taille normale; si c'est une vidéo, elle sera affichée dans un lecteur vidéo. Ce bouton fait appel à la méthode 'visualiser()' de la Source correspondante. Il y aura un exemple de cette visualisation dans les screenshots concernant l'Endoscope Virtuel.

### 7.5.2 L'onglet 'Panel Envois'

Illustration 12: L'outil PanelEnvois

1) Le panier: Concrètement le panier est une galerie qui stocke tout les fichiers envoyés par les autres outils de l'interface. Il propose aussi des fonctionnalités pour manager les fichiers: désélectionner tout, sélectionner tout, enlever la sélection du panier, visualiser, vider le panier et enregistrer sur le disque(pour enregistrer le fichier sélectionné sur le disque dur ).

2) Les images envoyées: Des icônes représentant les fichiers envoyés dans le panier. Il est à noter que lorsqu'un fichier est envoyé dans le panier, il est en fait copié et donc il est toujours présent dans l'outil d'origine.

3) Les informations sur le patient: c'est ici que sont affichées les données du patient, on peut les importer du réseau DICOM grâce au bouton 'import examen from DICOM', les créer sois-même avec le bouton 'create new examen' et en validant les changements.

4) La destination: permet de sélectionner le serveur sur lequel on va envoyer les fichiers sélectionnés dans la galerie du panier; ces fichiers seront rattachés aux 'informations patient' affichées en haut , le tout sera 'dicomisé' puis, envoyé sur le réseau DICOM.

### 7.5.3 Autre exemple, le Microscope Virtuel

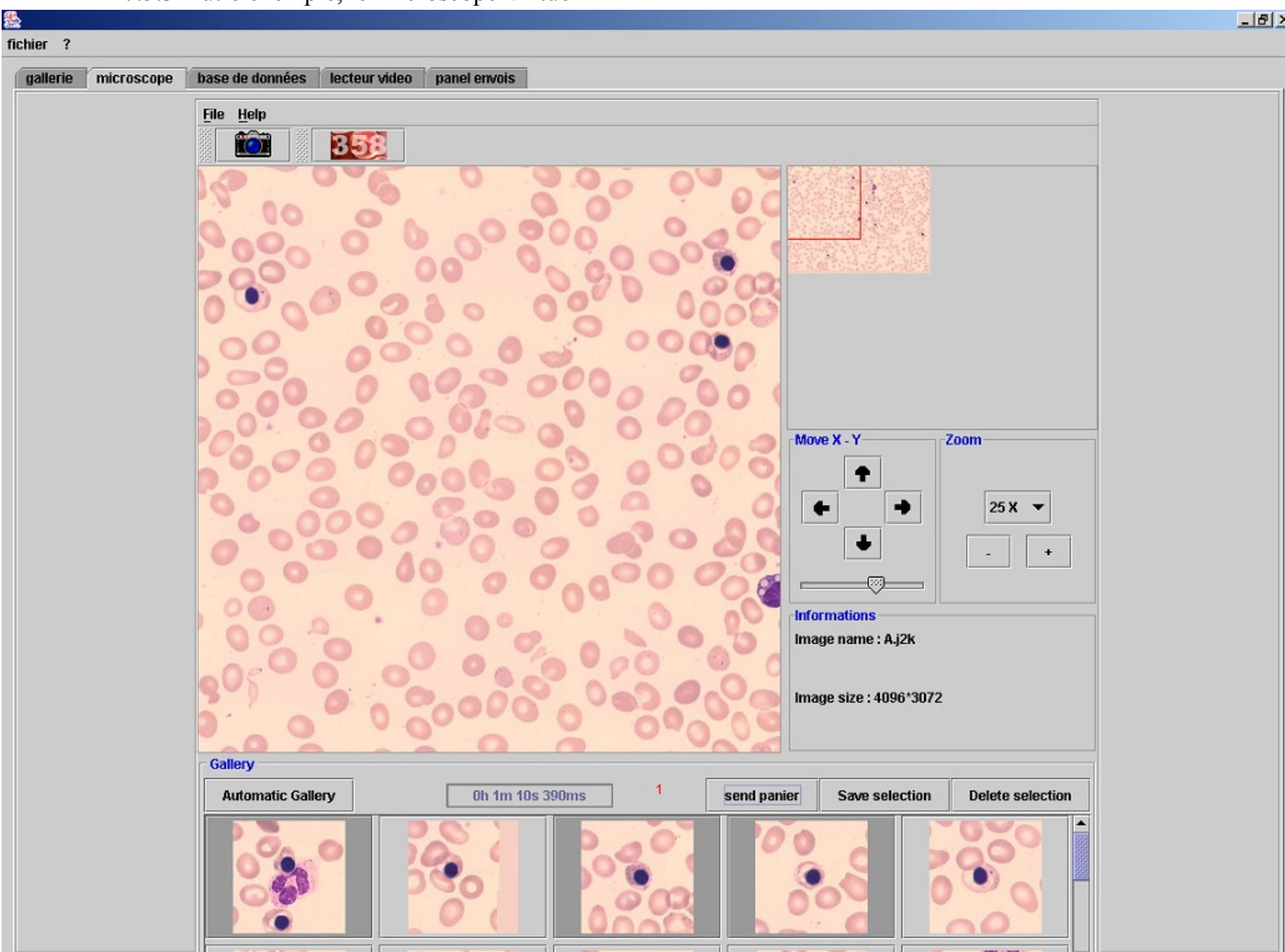


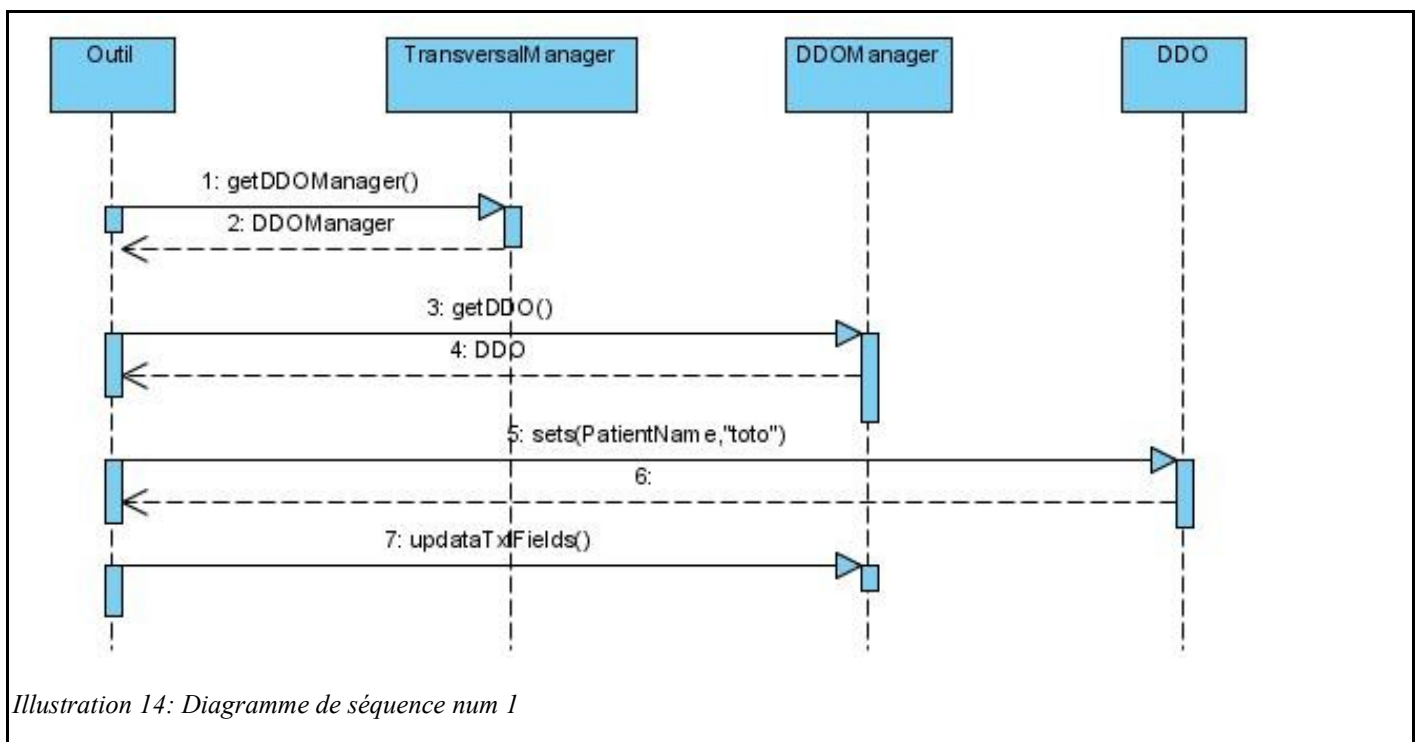
Illustration 13: Le Microscope Virtuel

Ce screenshot illustre l'intégration de l'outil 'Microscope Virtuel', nous pouvons voir en 1) que le bouton 'send panier' est présent et permet d'envoyer la sélection vers le panier.

## 7.6 Diagrammes de séquence

Afin de mieux comprendre l'interaction entre les classes, voici deux diagrammes de séquençement exposant l'ordre dans lequel sont appelés les classes dans les cas où, premièrement, un outil d'imagerie accède aux informations du patient et deuxièmement, un outil envoie dans le panier une image. Ces schémas sont exposés après les screenshots de façon à permettre au lecteur une meilleure compréhension des résultats donnés par la série d'interaction illustrée dans les diagrammes.

### 7.6.1 Diagramme de séquence pour accéder aux informations contextuelles



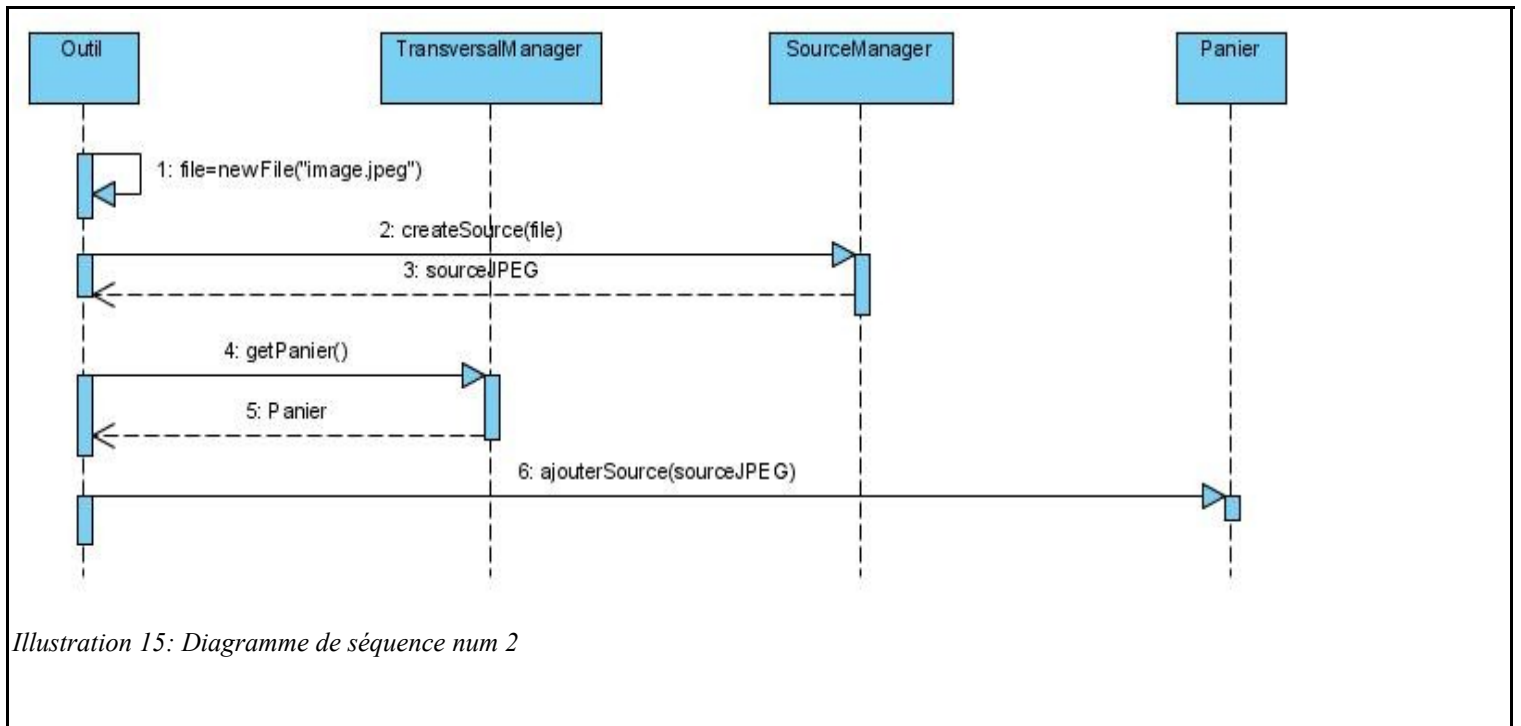
### 7.6.2 Explication du diagramme

- 1) L'Outil demande au TransversalManager (qu'il contient) de lui renvoyer le DDOManager
- 2) Le TransversalManager renvoie le DDOManager à l'Outil
- 3) L'Outil qui a maintenant une référence vers le DDOManager lui demande de lui donner le DDO (qui contient les données contextuelles)
- 4) Le DDOManager lui renvoie le DDO
- 5) L'Outil demande au DDO de mettre une par exemple, valeur "toto" dans le champs PatientName du DDO
- 6) Le DDO ne renvoie rien
- 7) L'Outil demande au DDOManager d'afficher à l'écran le contenu du DDO (cela affichera à



l'écran "toto" dans le champ prévu pour le nom du patient).

### 7.6.3 Diagramme de séquence pour envoyer un fichier dans le panier



### 7.6.4 Explication du diagramme

- 1) L'Outil crée un objet du type File (un fichier)
- 2) L'outil demande au SourceManager de lui renvoyer un objet du type Source en fonction de l'objet file qui est en paramètre
- 3) Le SourceManager lui renvoie l'objet sourceJPEG(puisque l'objet file était un fichier JPEG)
- 4) L'Outil demande au TransversalManager de lui donner le Panier
- 5) Le TransversalManager renvoie le Panier
- 6) L'outil demande au Panier d'ajouter l'objet SourceJPEG (cela à pour effet d'afficher dans l'outil PanelEnvois une nouvelle icône représentant SourceJPEG)

## 7.7 Conclusion

Les différents outils d'imageries développés au service de recherche informatique sont maintenant rassemblés dans une seule et même application. Ils peuvent entre-autre envoyer les images qu'ils ont générés sur le réseau interne DICOM de l'hôpital. La conception réalisée permet l'ajout facile d'un nouvel outil ou de nouvelles fonctionnalités.

Cette application peut encore être améliorée, la partie Discussion présentera de manière plus précise les objectifs atteints ainsi que des possibilités d'amélioration.



## Chapitre 8 Le prototype de lecteur vidéo

Cette partie expose les résultats concernant le prototype Endoscope Virtuel aussi appelé lecteur vidéo dû à sa polyvalence. Cet outil est représenté par la classe 'Lecteur' dans l'application d'intégration. Une modélisation par diagramme de classes ainsi qu'une explication sur son contenu compose les deux premières parties de ce chapitre. Il s'en suit une série de 'snapshots' pour montrer visuellement le résultat obtenu.

### 8.1 La conception, le diagramme de classe

Comme pour l'application d'interface commune, une conception en diagramme de classe a été réalisée, c'est cette modélisation qui est présentée dans la figure qui suit:

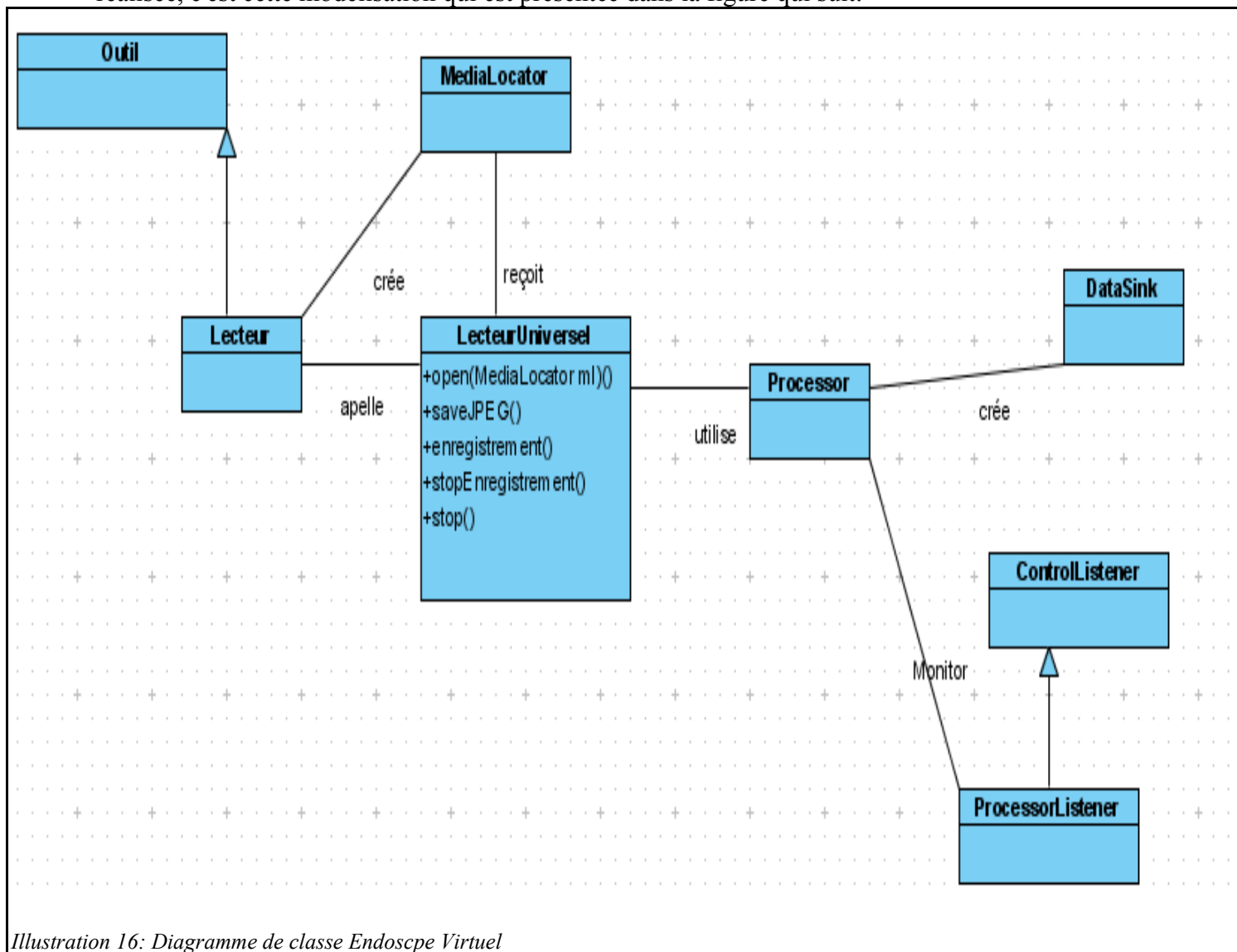


Illustration 16: Diagramme de classe Endoscope Virtuel

### 8.2 Explication des classes

Comme pour la première application, un schéma ne suffit pas à lui seul pour faire comprendre le rôle des classes du diagramme, voici donc une explication de celles-ci:

Les classes Outil et Lecteur sont déjà exposées dans le chapitre précédent voir le point 8.2

## LecteurUniversel

Classe représentant un lecteur vidéo qui peut être utilisé soit pour lire un fichier stocké sur le disque, soit de la vidéo provenant d'un outil de capture reconnu par JMF.

Elle fait appelle à la classe JMF 'Processor' pour avoir les fonctionnalités de base ( lecture, pause, stop, ....) mais aussi, pour accéder au contenu de la vidéo.

Elle peut aussi faire appelle à un 'Processor' particulier pour enregistrer une séquence vidéo.

## Processor

Classe JMF qui fournit l'affichage de la vidéo ainsi que certaines fonctionnalités de base pour la lecture de fichiers stockés sur disque(arrêter la lecture, pause, retour,...) .

## DataSink

Classe JMF utilisée pour écrire des données sur disque. Par exemple, lorsque l'on enregistre une vidéo provenant d'un appareil de capture, c'est un objet de cette classe qui s'occupe de prendre le flux vidéo en temps réel, de créer un fichier sur le disque puis, de déposer le flux à l'intérieur de celui-ci.

## ProcessorListener

Classe qui s'occupe de gérer la transition entre les différents états d'un Processor. Il implémente la classe JMF ControllerListener.

## ControllerListener

Classe JMF contenant une série de méthode à implémenter pour gérer les différents états d'un Processor

## MediaLocator

Classe qui renferme une référence vers le média à traiter.

## 8.3 Screenshots de l' application :

Voici une série de screenshots avec, pour certains éléments de l'image, jugés pas assez explicites , des commentaires . Ces screenshots illustrent l'application vidéo avec comme source d'entrée la webcam, le système 'Pinnacle' puis, la lecture d'un fichier stocké sur le disque dur.

### 8.3.1 Le lecteur vidéo intégré dans l' Interface, avec la webcam en entrée

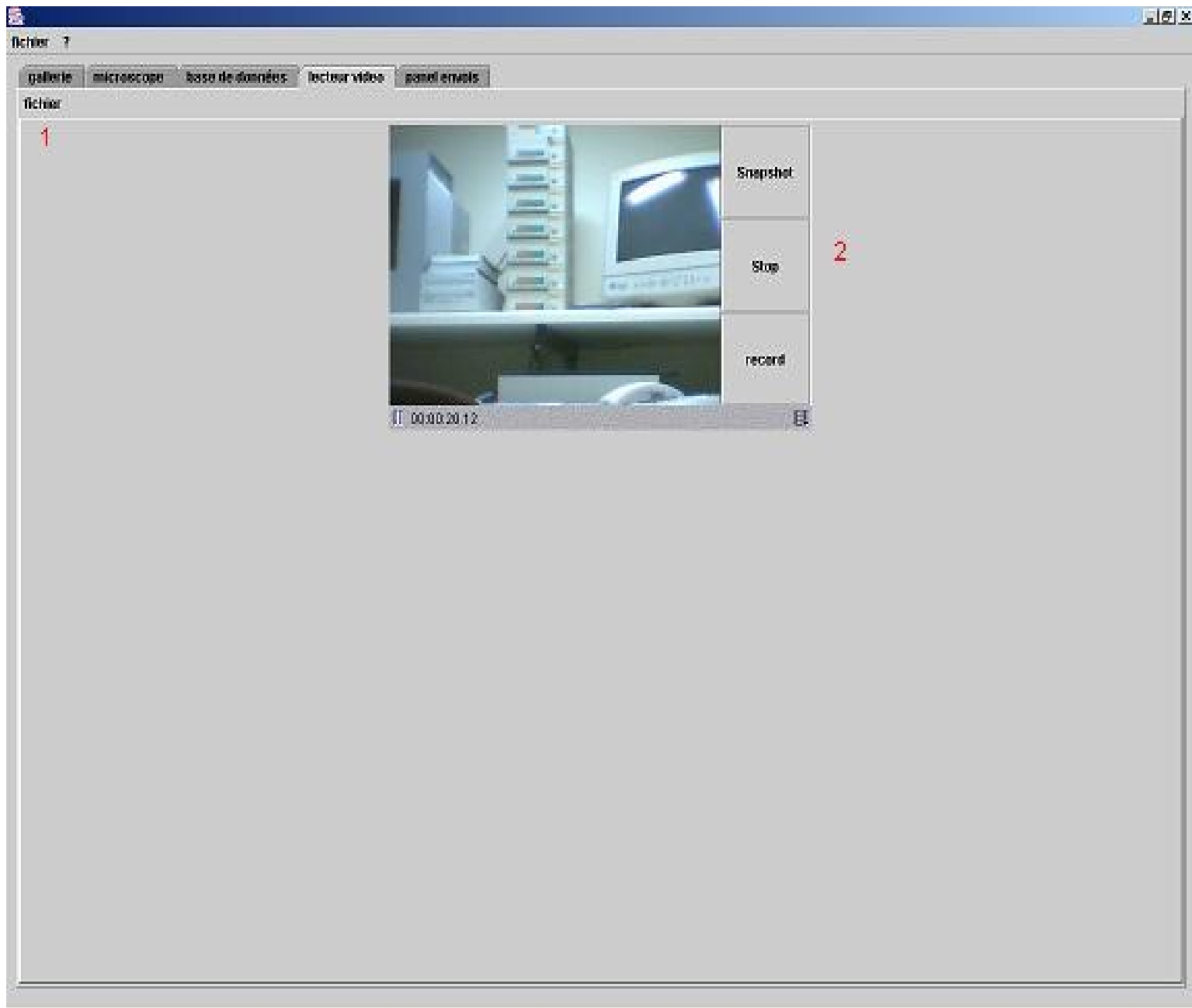


Illustration 17: Lecteur vidéo avec webcam

1) Le menu 'fichier': Il permet de sélectionner soit un fichier vidéo stocké sur le disque, soit l'appareil qui est enregistré dans JMF.

2) Les différents boutons:

Snapshot : Permet de prendre une capture 'image' sous le format JPEG de la vidéo affichée, au moment où on appuie sur le bouton.

Stop: arrête l'affichage de la vidéo en cours.

Record : enregistre le flux vidéo en cours de diffusion, après avoir cliqué dessus, le texte du bouton devient 'Stop record' et, lorsque l'on reclique dessus, on arrête l'enregistrement.

Remarque: Tout les fichiers capturés avec cet outil sont directement envoyés dans le panier.

### 8.3.2 Le panier avec un fichier de capture vidéo

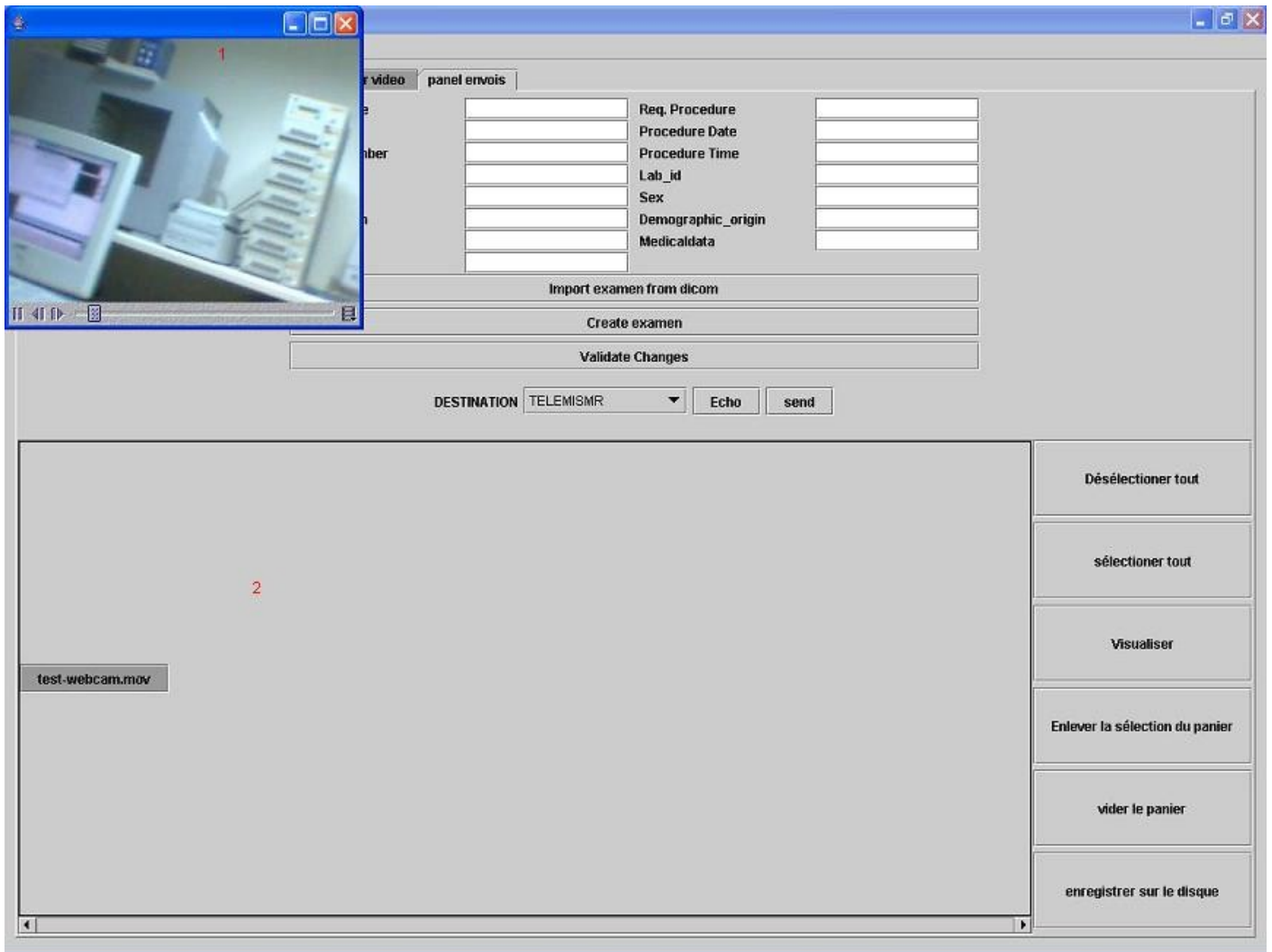


Illustration 18: Panier avec la lecture d'un fichier vidéo

- 1) La vidéo contenue dans le fichier vidéo sélectionné. Le lecteur a été créé ,via le bouton 'visualiser' , qui fait appel à la méthode 'visualiser()' de la classe SourceVideo.
- 2) Le Panier avec les fichiers en provenance du lecteur vidéo

### 8.3.3 Le lecteur avec en entrée le système Pinnacle et le magnétoscope

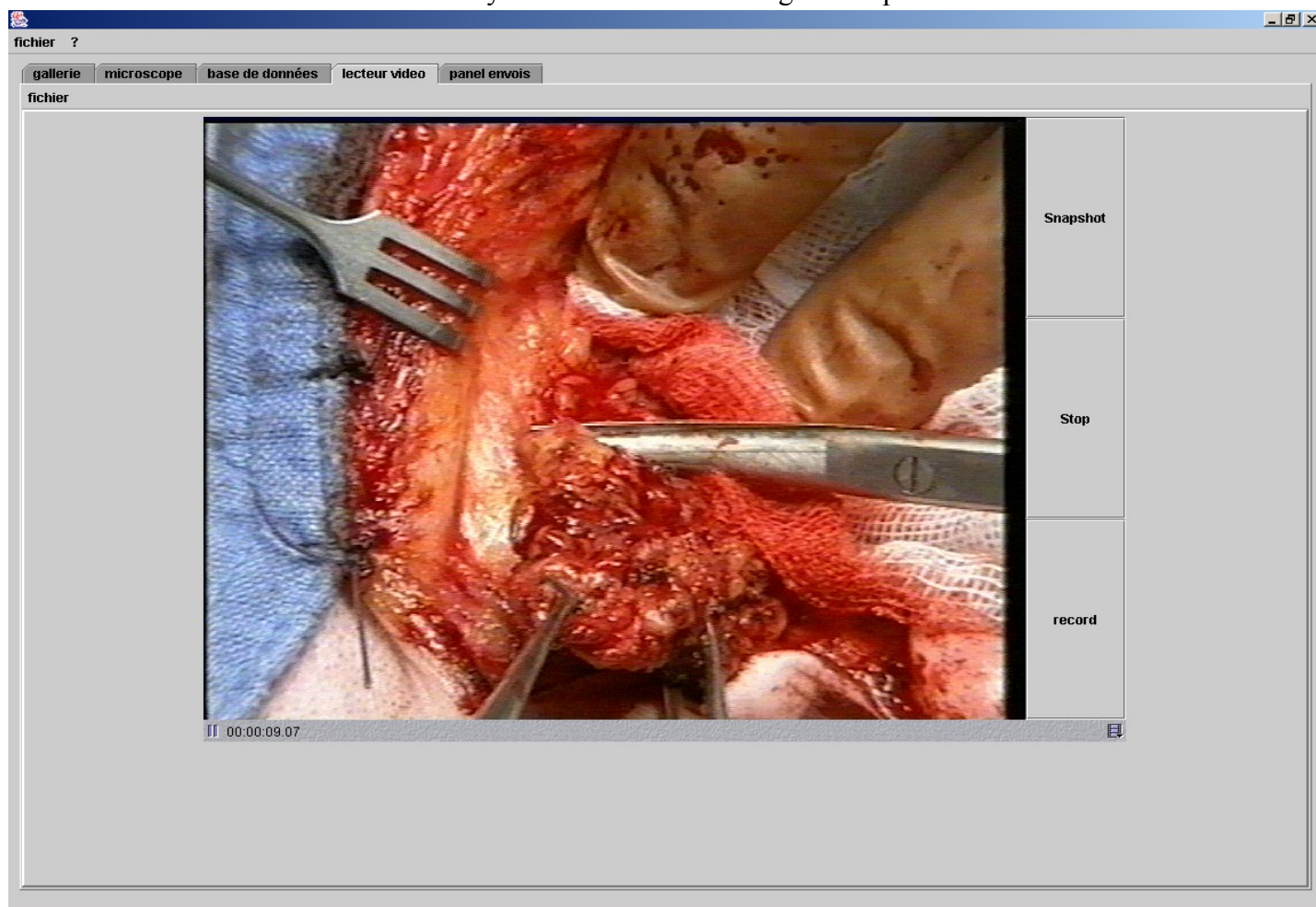


Illustration 19: Lecteur avec Magnétoscope

### 8.3.4 Lecture d'un fichier stocké sur le disque dur

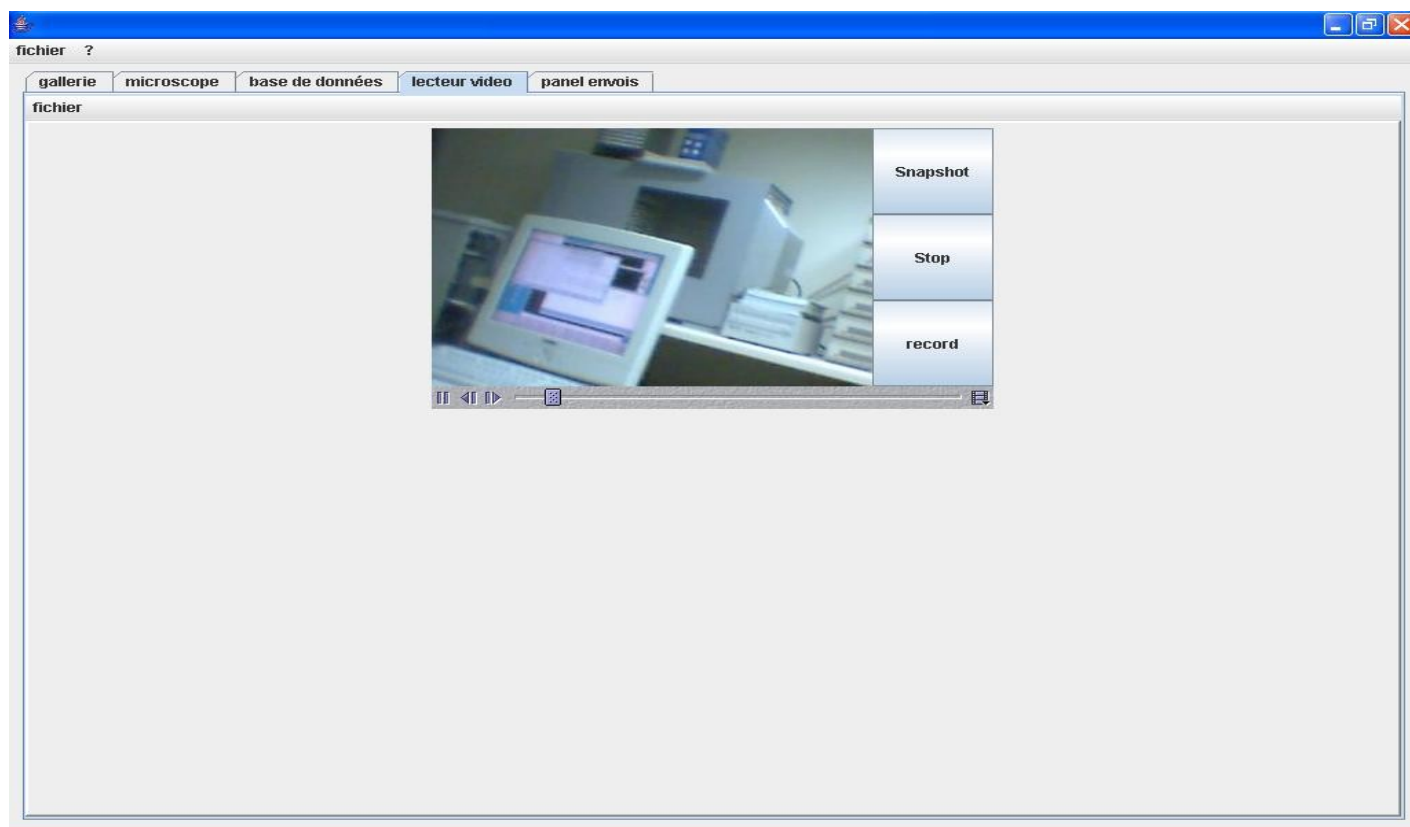


Illustration 20: Lecteur avec fichier

## **8.4 Conclusion**

Le service de recherche informatique dispose donc maintenant d'un outil de lecture vidéo complètement intégré dans l'Interface Commune et entièrement codé dans le langage JAVA. Cet outil polyvalent permet de visionner un flux vidéo provenant d'un appareil connecté aussi bien sur un port usb que sur un port firewire. Il permet aussi à l'heure actuelle d'enregistrer une séquence vidéo au format QuickTime ou encore, de capturer des images au format JPEG. Ce premier prototype est donc concluant et nous permet d'envisager à l'avenir de nouvelles fonctionnalités. Certaines pistes d'évolutions seront présentées dans la partie Discussion.

# Chapitre 9 Etude de la norme MPEG-2

Cette étude est basée sur les références [MPEG], [MPEGT1],[MPEGT2] et [MPEGT3]

## 9.1 Introduction

Aujourd'hui, le monde de la vidéo numérique est en pleine extension, nous pouvons voir arriver sur le marché de nouveaux services proposés aux consommateurs :la télévision numérique via internet, la télévision interactive qui permet à l'utilisateur de choisir son programme télé personnalisé, l'achat de films directement par internet,... En effet, l'industrie de la vidéo sait utiliser les nouvelles technologies telles internet ou, les nouveaux supports média comme le blue-Ray de sony afin d'attirer le consommateur et lui proposer l'achat de nouveaux produits.

Il va de soit que la multitude croissante des acteurs (producteur, opérateurs propriétaire de contenu,distributeur,...) du milieu de la vidéo numérique implique une complexité plus grande dans la gestions des données numérisées, que ce soit pour le transport, le stockage ou la diffusion sur réseau. Cette complexité nécessite la création de normes internationales afin de permettre aux différents protagonistes de communiquer entre eux , tout en limitant la complexité pour le consommateur qui pourra par exemple, lire une vidéo provenant d'un distributeur américain ou d'un distributeur européen sans difficulté supplémentaire.

Lorsque l'on parle de stockage vidéo ou de diffusion vidéo à travers réseaux, il se pose un autre problème, c'est la taille des données numérisées. En effet, si l'on souhaite diffuser en direct sur internet une émission ou un journal, il faut limiter la consommation en bande passante de la vidéo afin, que l'utilisateur final soit aussi à l'aise devant son ordinateur que devant sa télévision traditionnelle. Cela implique un codage du fichier vidéo original afin de compresser les données et de réduire leurs tailles.

C'est pour répondre à ces deux problèmes de standardisation et de compression numérique qu'a été créée la norme MPEG-2, ce chapitre présente une étude de celle-ci et, à travers elle, une première approche de concepts souvent utilisés dans la sphère vidéo numérique.

## 9.2 MPEG-2 Présentation

### 9.2.1 Présentation du MPEG-2

Développée par le Moving Picture Experts Group, le MPEG-2 est la norme de seconde génération (1994) du qui fait suite à MPEG-1. Cette norme définit les aspects compression de l'image et du son et le transport à travers des réseaux pour la télévision numérique.

Ce format vidéo est actuellement utilisé pour les DVD et SVCD avec différentes résolutions d'image mais aussi, dans la diffusion de télévision numérique par satellite, câble, réseau de télécommunications ou hertzien .Même si le principe a été posé en 1994, le mpeg2 est encore d'actualité, c'est le format audio et vidéo qui sera utilisé sur le système Blu-Ray de Sony mais aussi, avec le MPEG-4 pour le HD-DVD . Pour rappel, le Blu-Ray et le HD-DVD sont les successeurs du dvd .

Le MPEG-2 n'est pas monopolisé par l'industrie de la consommation vidéo, c'est aussi le format le plus utilisé dans le monde de l'imagerie médicale. De ce fait, la norme DICOM (voir [DICOM]) propose un supplément qui permet à l'informaticien de dicomiser la vidéo codée au format MPEG-2, c'est le supplément n°42 'MPEG2 Transfert Syntax'.

### 9.2.2 La notion de codec

Le codec est un mot passe-partout et des erreurs sont souvent commises. Tout d'abord, il ne faut pas confondre codec et norme ou format de données. La norme définit la manière de traiter les données, le codec implémente cette manière. On dit que le codec implémente la norme. Codec signifie 'Codeur-Décodeur' et désigne un procédé capable de compresser ou de décompresser un signal, analogique ou numérique, en un format de données. Ce procédé peut exister sous forme matérielle (carte graphique qui réalise ce procédé sans aide logicielle) ou logicielle. Les codecs encodent des flux ou des signaux pour la transmission, le stockage ou le cryptage de données. D'un autre côté, ils décodent ces flux ou signaux pour l'édition ou le visionnage. Le but premier des codecs est de pouvoir traiter un maximum de données avec un minimum de ressources. Ils sont utilisés dans toutes sortes d'application, la téléphonie, les vidéos-conférence, la télévision numérique,...

Il existe une multitude de codecs, mais ils peuvent être divisés en deux grandes familles suivant leur manière de compresser les données. En effet, on distingue deux types de compression. La compression non-destructive (on dit aussi sans pertes ou lossless) qui permet de retrouver le signal initial tel qu'il était avant codage et la compression destructive (avec pertes ou lossy) qui prend en compte les caractéristiques des données à compresser et qui peuvent retirer des informations pouvant être considérées comme non pertinentes. Par exemple, les informations non perçues par l'humain sont retirées des formats JPEG et Vorbis.

### 9.3 La norme MPEG-2

La norme est développée par le Moving Picture Experts Group, en partenariat avec l'industrie de la vidéo. Voici une vue générale de la composition de ce standard et ce qu'il propose à l'heure actuelle.

**La première partie** MPEG-2 présente le flux élémentaire de données vidéos et audios ainsi que la combinaison de ceux-ci. Ce flux est spécifié de deux façons : le flux-programme et le flux-transport, chacun d'eux est optimisé pour différents types d'applications.

**La deuxième partie** MPEG-2 est construite à partir des capacités de compressions du MPEG-1, pour offrir une large palette d'outils d'encodages. Ces outils sont regroupés en Profiles pour offrir différentes fonctionnalités.

**La troisième partie** MPEG-2 expose un multi-channel audio qui est compatible avec la norme audio du MPEG-1.

**Les parties 4 et 5** MPEG-2 correspondent aux parties 4 et 5 du MPEG-1.

**La sixième partie** MPEG-2 - Digital Storage Media Command and Control (DSM-CC) est la spécification d'un ensemble de protocoles qui fournissent les fonctions de contrôle et les opérations spécifiques pour gérer les bitstreams MPEG 1 et 2.

**La septième partie** MPEG-2 est la spécification d'un algorithme de codage d'un multichannel audio qui n'est pas contraint d'être compatible avec la norme audio du MPEG-1.(partie toujours en cours).

**La huitième partie** MPEG-2 partie abandonnée concernant un codage de données pour un input de 10 bits

**La neuvième partie** MPEG-2 est la spécification du Real-time Interface (RTI) pour transporter des décodeurs de flux qui peuvent être utilisés pour l'adaptation à tout les réseaux transportant des flux de transport



**La dixième partie** MPEG-2 est le test de conformité du DSM-CC (toujours en développement).

Il est à noter que certaines parties sont toujours en cours et d'autres ont été abandonnées, jugées peu intéressantes par l'industrie du film. Ce standard est un 'on-going processus', c'est à dire qu'il est constamment sujet de modifications et d'ajouts, en fonction des besoins exprimés par les utilisateurs.

#### **9.4 Les fonctionnalités**

MPEG-2 propose une série de fonctionnalités qui le rend particulièrement intéressant pour les services de vidéos numériques:

Premièrement, la compression vidéo MPEG-2 est toujours compatible avec le standard précédent MPEG-1, cela signifie qu'un décodeur MPEG-1 peut décoder un flux MPEG-2 même s'il n'aura pas accès à toutes les fonctionnalités attachées à ce flux.

Deuxièmement, il propose plusieurs modes d'enregistrement audio en fonctions des besoins , cela comprend notamment la qualité mono, stéréo, high quality ainsi que d'autres fonctionnalités particulières.

Troisièmement, il permet le transport multiplexing c'est à dire, qu'il est possible de combiner différents flux MPEG-2 en une seule transmission. Cela est particulièrement intéressant pour la diffusion à travers un réseau ou internet.

Quatrièmement, il est possible d'attacher avec le flux vidéo toute une série d'informations privées telles que le propriétaire du contenu, les droits associés avec des informations permettant de chiffrer les données afin de ne permettre le visionnage qu'à certains types d'utilisateurs. Cela intéresse évidemment les distributeurs de vidéos numériques qui ne souhaite pas que le contenu de la vidéo soit accessible à tout le monde, mais seulement aux abonnés par exemple.

#### **9.5 Profile and Level**

Le MPEG-2 a été conçu pour supporter un large panel d'applications et de services, la norme propose donc d'implémenter en fonction de ses besoins le bit-rate(quantité de bit par seconde), la résolution et la qualité. Le standard définit 4 Profiles et 4 Levels pour assurer l'inter-opérabilité entre les applications.

Le Profile définit la résolution de l'espace couleur et, la scalabilité du bitstream. Le Level définit quand à lui, le minimum et le maximum pour la résolution de l'image, la luminance, le nombre de couches vidéo et audio supportées et, le nombre maximum de bit rate par Profile .

Ces Levels et Profiles sont décrits dans les tableaux suivant ainsi que les combinaisons reconnues par le standard :

<b>Level</b>	<b>Profile</b>	<b>Simple</b>	<b>Main</b>	<b>SNR scalable</b>	<b>Spatial scalable</b>	<b>High</b>	<b>Multiview</b>	<b>4: 2 :2</b>
<b>High</b>		illégale	ATSC 1080i, 720p60, HD-DVB (HDTV)			reconnu	illégale	Potentiellement pour la HD de Panasonic
<b>High 1440</b>		illégale	HDV			reconnu		Potentiellement pour la HD de sony
<b>Main</b>		PDA's	DVD, SD-DVB	avec " SNR scalability "	" Main " avec " spatial scalability "	reconnu	reconnu	4 :2 :2
<b>Low</b>		Petits appareils sans fils	Set-top boxes (STB)	" Main " avec " scalability "				illégale

*Illustration 21: Combinaison Levels et Profiles*

Combinaisons des Profiles et Levels reconnus par le standard avec pour certains, un commentaire.

<b>Level</b>	<b>Résolution</b>	<b>Pixels/seconde</b>	<b>Débit maximum</b>	<b>Notes</b>
Low	352 x 240 x 30	3.05 Millions	4 Mb/s	CIF, équivalent au VHS
Main	704 x 480 x 30	10.40 Millions	15 Mb/s	CCIR 601, studio TV
High 1440	1440 x 1152 x 30	47.00 Millions	60 Mb/s	HDTV
High	1920 x 1080 x 30	62.70 Millions	80 Mb/s	production vidéo, standard SMPTE 240M

Note : Le niveau " Low " est similaire à MPEG-1.

*Illustration 22: Les différents Levels de la norme MPEG-2*

Profile	Commentaires
Simple	identique à " Main " mais sans les images du type B, prévu pour les applications logiciels.
Main	TV par câble, satellite , 95% des utilisateurs
SNR scalable	identique à " Main " mais avec " SNR scalability extensibility "
Spatial scalable	identique à " Main " mais avec " Spatial scalability extensibility "
High	
Multiview	
4:2:2	" Main+ " avec un codage 4:2:2

Illustration 23: Les différents profiles de la norme MPEG-2

Remarque: La combinaison d'un Profile et d'un Level est exprimée de cette façon :

'nom du Profile' @ 'nom du Level' par exemple, MP@ML pour la combinaison de Main Profile et Main Level (qualité DVD).

## 9.6 Le flux de données

Ce point décrit la composition d'un flux de donnée MPEG-2. En présentant une figure qui illustre la hiérarchie des éléments d'un flux vidéo MPEG-2. Elle sera suivie par une description plus détaillée de chaque élément encadré en gras. Les autres éléments étant des paramètres, ils ne seront pas détaillés.

### 9.6.1 Hiérarchie du flux vidéo MPEG-2:

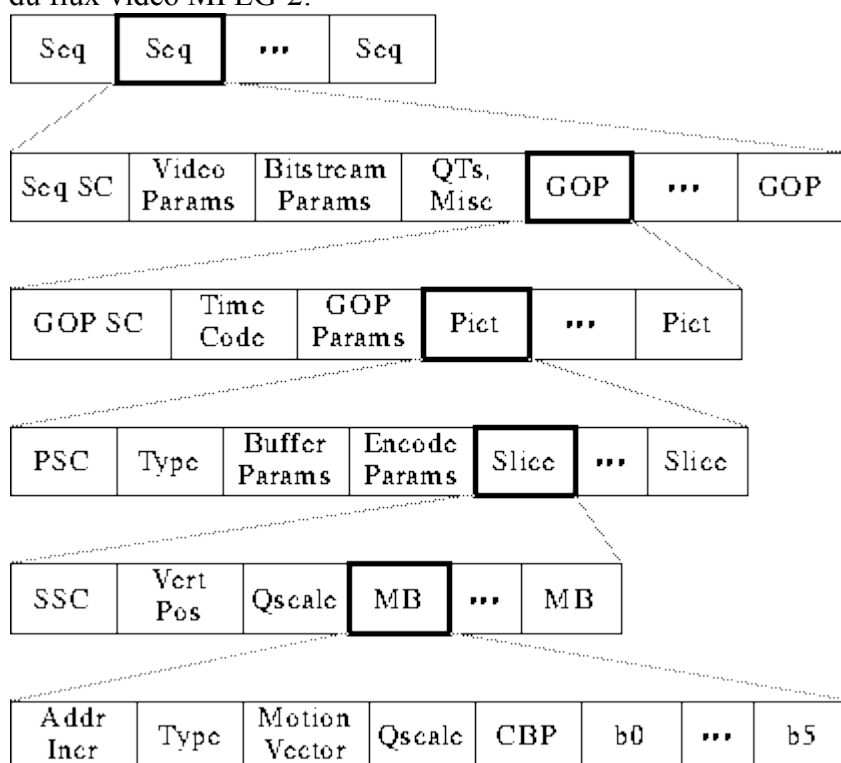


Illustration 24: La hiérarchie du flux MPEG-2

Explication des différents niveaux:

### Séquence vidéo

Commence avec un séquence header (qui peut contenir des headers additionnels), inclus un ou plusieurs groupes d'images (GOP) et, se termine par un code de fin de séquence.

### Group of Pictures (GOP)= groupe d'images

Contient un header suivit d'une série de une ou plusieurs images, qui permettent l'accès aléatoire dans la séquence.

### Picture

C'est l'unité primaire de codage dans une séquence vidéo. Une image est en fait digitalisée en étant transformée en une matrice de pixels, chaque pixel est représenté par un nombre de luminance (densité de la lumière) et deux nombres de chrominance (information sur la couleur). Une picture est donc représentée initialement par trois matrices rectangulaires représentant la luminance(Y) et deux valeurs de chrominances (Cb et Cr).

Une technique pour réduire le taux de données est de supprimer une partie des données sur la chrominance, de façon à ce que ce ne soit pas perceptible par l'oeil humain. Les données restantes représentent alors plusieurs pixels à la fois (perte d'information). On garde l'ensemble des données sur la luminance car l'oeil humain y est plus sensible.

### Slice

Se compose de un ou plusieurs macroblocks. L'ordre des macroblocks à l'intérieur d'une slice est de gauche à droite puis de haut en bas. Les slices sont importants dans la gestion des erreurs. Si le bitstream contient une erreur, le décodeur peut sauter le début de la prochaine slice. Avoir plus de slice dans le bitstream permet de mieux faire face aux erreurs cependant, on utilise alors des bits qui autrement auraient pu être utilisés pour stocker de l'information sur l'image, pour, par exemple améliorer la qualité de celle-ci.

### Macroblock

C'est l'unité de base dans l'algorithme de codage MPEG. Chaque macroblock représente dans une image un morceau de 16x16 pixels. Un macroblock est composé d'une matrice 16x16 pour la luminance Y (4 blocks) plus un nombre variable de blocks pour la chrominance (Cr et Cb). Ce nombre varie en fonction du Level et du Profile et est indiqué dans le header de la séquence. Comme vu précédemment, on ne peut pas garder toute l'information concernant la chrominance (Cb et Cr). Plus une vidéo sera de qualité, plus elle conservera l'ensemble des données initiales, et comportera donc 4 blocks de chrominance (on ne peut aller au-delà de 2 blocks Cb + 2 blocks Cr).

### Block

La plus petite unité de codage dans l'algorithme MPEG. Il consiste en une matrice de 8x8 pixels et peut être d'un des trois types suivant, luminance(Y), red chrominance ou blue chrominance.

Le block est aussi l'unité de base dans le codage intra-frame.

Voici une figure représentant un macroblock composé de 6 blocks:

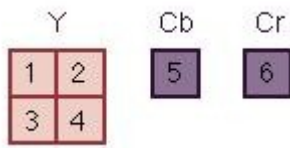


Illustration 25: Macroblock

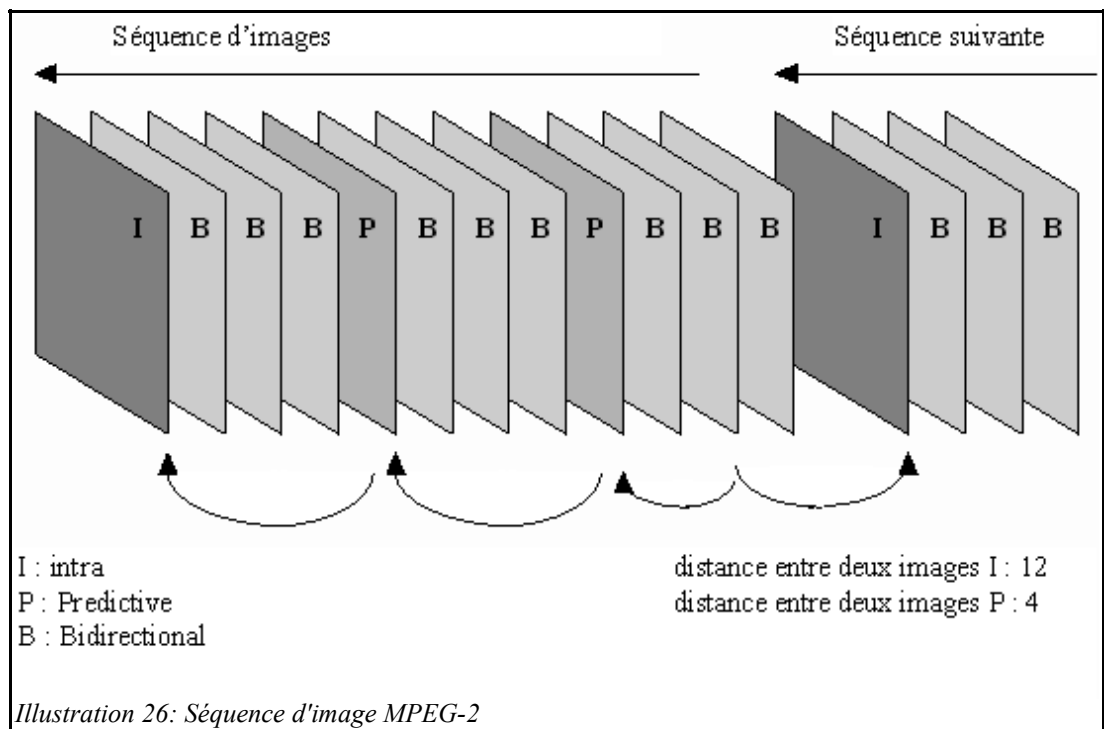
### 9.6.2 Picture Types

Voici une explication des différentes images contenue dans un flux vidéo:

Le standard MPEG définit spécifiquement trois types d'images:

1. Intra Pictures (I-Pictures)
2. Predicted Pictures (P-Pictures)
3. Bidirectional Pictures (B-Pictures)

Ces trois types d'images sont combinées pour former un GOP. Voir figure suivante.



#### Intra Pictures

Appelées aussi I-Picture, sont codées en utilisant uniquement l'information présente dans l'image elle-même et, fournit des points potentiels d'accès aléatoires dans les données vidéo compressées. Elles utilisent seulement un codage de transformation ce qui donne une compression modérée. Typiquement, elle utilise 2 bits par pixel codé.

#### Predicted Pictures

Appelées aussi P-pictures, sont codées en fonction de la plus proche I ou P picture, cette technique s'appelle 'forward prediction' et est illustrée dans la figure au-dessus. Comme les I-pictures, les P-pictures peuvent aussi servir de référence pour des B ou P-pictures. Elles utilisent la technique de 'motion compensation', pour fournir une plus grande compression que les I-Picture.

## Bidirectional Pictures

Appelées aussi B-Pictures, sont des images qui utilisent les images passées et futures comme référence pour le codage. Cette technique est appelée 'bidirectional prediction'. La B\_Picture fournit la meilleur compression seulement, le temps de calcul pour le codage est le plus long.

Ces trois types d'image correspondent en fait à trois mode d'opération que propose la norme, en fonction de la nécessité d'un encodage rapide ou de la nécessité d'une compression plus optimale, l'implémenteur pourra choisir d'utiliser soit les I-Picture , les I-Picture + les P-Picture ou, les trois à la fois.

## 9.7 L'algorithme de codage

### 9.7.1 Codage des I-picture

Voici l'algorithme MPEG-2 d'encodage intra-frame (codage d'une image sans faire appel à une autre image)et pour chaque étape du processus , une explication afin de permettre de comprendre les concepts de base de la compression vidéo.

Il est à noté que pour décoder , il suffit d'exécuter l'algorithme en sens inverse.

#### 9.7.1.2 Le schéma d'encodage

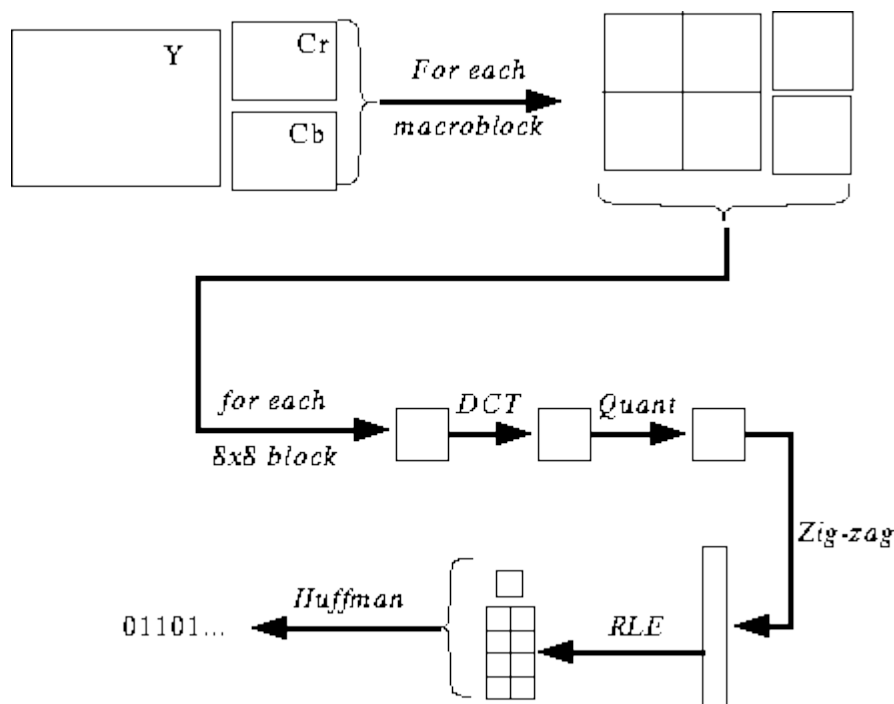


Illustration 27: Algorithme d'encodage MPEG-2

### 9.7.2 Explication des différents étapes:

#### 9.7.2.1 Discete cosine transform (DCT)

La DCT est une opération mathématique qui transforme un ensemble de données d'un domaine

spatial et les transforme en une représentation équivalente dans le domaine fréquentiel. Le nombre de données doit être fini et, pour un temps de calcul optimal puissance de deux. Rappelons que ce processus est appliqué à une matrice 8x8 (8 étant puissance de 2).

C'est un codage sans perte, le processus inverse produit à l'identique les informations initiales. Ce procédé a pour résultat de concentrer fortement l'information, afin d'exécuter par la suite de manière plus efficace des compressions avec perte sur l'ensemble des données.

### 9.7.2.2 La Quantification

L'oeil humain est très fort pour voir une différence de luminosité sur de larges surfaces ou entre deux contrastes très différents. Cependant, il distingue difficilement ou pas du tout les variations de luminosités entre les fréquences qui sont hautes, la quantification va permettre d'atténuer les fréquences irrémédiablement. Cela permet une réduction avec perte de l'information contenue dans une image, qui peut être élevée pour les composants avec les plus fortes fréquences. Ce procédé de compression s'appelle la quantification.

Ce processus est appliqué à la matrice résultante du DCT (où chaque composant représente une fréquence), il consiste à diviser chaque composant du domaine de fréquence par une constante qui lui est propre puis, à arrondir à l'entier le plus proche. Il est illustré avec les matrices suivantes:

-415	-33	-58	35	58	-51	-15	-12
5	-34	49	18	27	1	-5	3
-46	14	80	-35	-50	19	7	-18
-53	21	34	-20	2	34	36	12
9	-2	9	-5	-32	-15	45	37
-8	15	-16	7	-8	11	4	7
19	-28	-2	-26	-2	7	-44	-21
18	25	-12	-44	35	48	-37	-3

La matrice de coefficients DCT  
(Domaine de fréquence)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	109

La matrice de quantification

-26	-3	-6	2	2	-1	0	0
0	-3	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

La matrice résultante de la quantification

Exemple de calcul d'un composant pour le premier nombre un haut à gauche de la matrice:

$-415/16 = -25,9375$ , on arrondit à -26

Remarque: nous pouvons voir ici une perte d'information en effet, même si ce processus est réversible, à chaque arrondi, on perd le coefficient DCT original et on le remplace par une approximation.

Même exemple que le précédant mais exécuté à l'inverse:

$-26 \times 16 = -416$  au lieu de  $-415$  d'où la perte d'information.

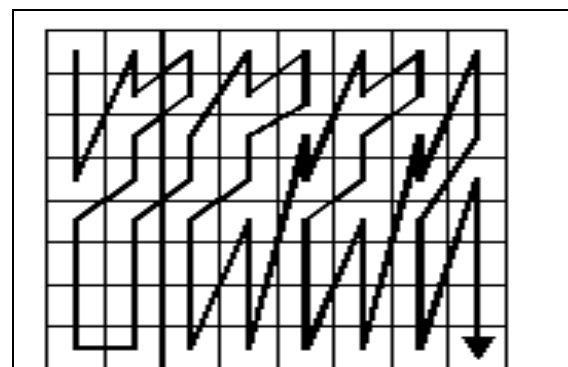
### 9.7.2.3 Ordonnancement ZigZag

Cette opération convertit un tableau de deux dimensions (la matrice résultante de la quantification) en une séquence de une dimension. On effectue le processus en insérant chaque case de la matrice dans un vecteur à une dimension selon un certain ordre pré-définis. Le standard mpeg-2 introduit un pattern à cet effet.

Sur la matrice  $8 \times 8$ , cela donne un vecteur de longueur 64 (car il y a 64 cases dans la matrice).

-22	17	1	1	0	0	0	0
-21	-15	1	0	0	0	0	0
11	3	-4	0	0	0	0	0
-3	2	2	0	0	0	0	0
0	-1	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Exemple de matrice résultante d'une quantification



C'est le pattern MPEG-2, il définit le chemin à suivre pour former le vecteur ZigZag à travers la matrice  $8 \times 8$ .

Vecteur zigzag, il a une longueur de 64 éléments puisqu'une matrice  $8 \times 8$  contient 64 cases:

$(-22, -21, 11, -3, 17, -15, 1, 1, 3, 2, 0, 0, 0, 0, 0, 0, -1, 2, -4, 1, 0, 0, 0, 0, 0, -1, 0, 0, 0, \dots)$  il n'y a plus que des zéros...

remarque: Pour faire le processus inverse, il suffit d'effectuer le chemin défini dans le pattern mais en sens inverse et en commençant par le dernier élément du vecteur ZigZag.

### 9.7.2.4 Run-length encoding

Méthode de compression qui consiste à repérer des répétitions consécutives puis à indiquer le nombre de répétitions. Dans le cas du MPEG-2, on enlève du vecteur ZigZag tout les zéros et, à la



place, on indique le nombre de zéro qu'il y avait à cet endroit dans un autre vecteur.

-22	-21	11	-3	17	-15	1	1	3	2	-1	2	-4	1	-1
Vecteur ZigZag transformé														

0	0	0	0	0	0	0	0	0	0	7	0	0	0	5
Vecteur-Run correspondant														

Remarque n°1: Le '7' dans la onzième case du vecteur-Run indique qu'il y avait sept zéros avant le '-1' correspondant dans le vecteur ZigZag transformé(onzième case).

Remarque n°2: Pour retrouver le vecteur ZigZag original(avec les zéros), il suffit de s'aider du vecteur-Run pour insérer les zéros au bon endroit. Par exemple, ajouter sept zéros avant le '-1' dans vecteur ZigZag transformé.

#### 9.7.2.5 Huffman Encoding

Le codage de Huffman est un algorithme de compression mis au point en 1952 par David Huffman. C'est une compression de type statistique utilisant une méthode 'd'arbre' permettant de coder les octets revenant le plus fréquemment avec une séquence de bits plus courte que pour les octets apparaissant plus rarement, d'où un gain de place dans le flux total d'octets. Voici un exemple de méthode à suivre pour effectuer un tel codage:

Première étape, calculer la probabilité de chaque symbole

symbole	A	B	C	D	E	F
probabilité	0,39	0,24	0,18	0,09	0,07	0,03

Deuxième étape, arranger les symboles en arbre. Aucun symbole de la table de départ n'a d'enfants. On commence par regrouper les deux symboles les moins probables en feuille et on leur donne comme parent un nouveau symbole fictif, celui qui a la plus petite probabilité est attaché au bit '1'.

Dans notre cas, E et F deviennent les enfants d'un nouveau symbole nommé M, on attache à E le bit '0' et à F le bit '1'.

On recommence le processus avec le symbole fictif et le symbole de la table de départ qui est le moins probable exceptés les symboles déjà utilisés (E et F). On recommence jusqu'à épuisement des symboles de la table de départ.

E=0,07(0)	M=0,10(0)	N=0,019(0)	O=0,37(0)	P=0,61(0)	
F=0,03(1)					
	D=0,09(1)				
		C=0,18(1)			
			B=0,24(1)		
				A=0,39(1)	

Voici l'arbre final

Symbole	A	B	C	D	E	F
Probabilité	0,39	0,24	0,18	0,09	0,07	0,03
code	1	01	001	0001	00000	00001

On assigne un code à chaque symbole en parcourant l'arbre

### 9.7.3 Codage des P-picture

Avant tout, voici une figure illustrant le procédé:

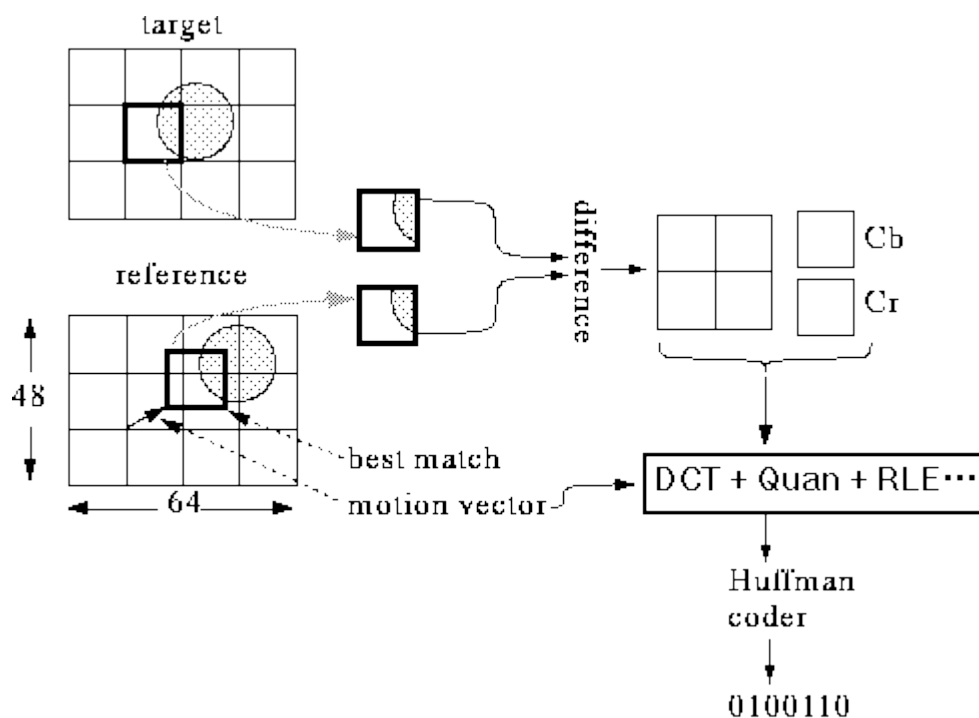


Illustration 28: Algorithme P-picture

Les P-picture sont codées en fonction d'une image I ou P que l'on appelle image de référence, l'image à coder est appelée l'image cible. A ces fins, on utilise la redondance temporelle, c'est à dire

le fait qu'une même information peut-être présente plusieurs fois dans le temps. Par exemple, un point rouge qui se promène sur l'écran lors de la vidéo, l'information contenue dans ce point est toujours la même, par contre le point change de place à chaque nouvelle image.

Dans ce processus, chaque macroblock est prédit à partir d'un macroblock de référence. Etant donné que le macroblock de référence n'est pas forcément à la même place que celui ciblé, il faut donc calculer la différence de position entre les deux , cette différence est appelée l' erreur de prédiction, c'est cette erreur qui est codé avec la technique du DCT. Ensuite, on continue le processus de codage intra-frame expliqué précédemment. Il y a donc ici une étape de calcul (la prédiction ) en plus que dans le codage des I-Pictures .

Il est à noté que la matrice de quantification utilisée pour les blocks d'erreurs de prédiction est différente de celle utilisée dans le cas de blocks classiques (blocks des I-picture)

#### 9.7.4 Codage des B-picture

Le codage des B-pictures s'effectue de la même manière que pour les P-picture sauf, qu'il y a deux images de références plutôt qu'une seule, une image passée et une image future. Ce processus nécessite donc encore plus de calcul que les P-picture.

Voici une image illustrant le procédé:

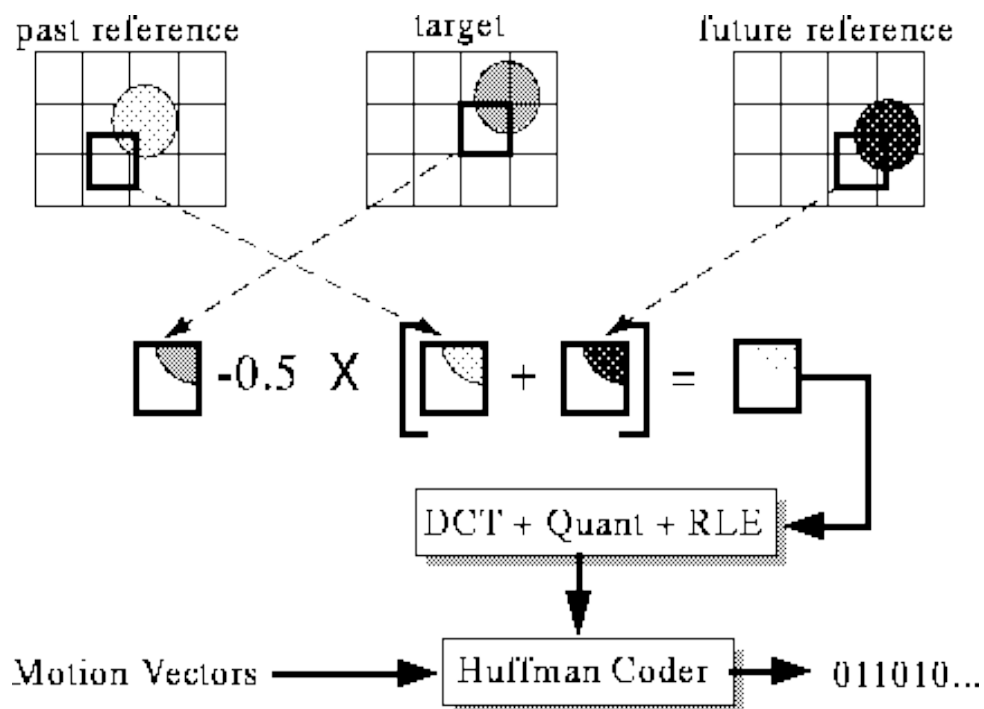


Illustration 29: Algorithme B-picture

## 9.8 Conclusion

Dans ce chapitre, nous avons vu la nécessité de disposer d'un standard de compression de données vidéos. Nous avons pour cela étudié le cas particulier de la norme MPEG-2 et, à travers son explication, nous avons découvert des notions de base souvent utilisées dans le milieu de la vidéo, que ce soit des concepts généraux tel que le codec, la compression avec ou sans pertes ou encore, des outils de compressions qui permettent une réduction conséquente de la taille des données.

Cette étude est particulièrement intéressante dans le cadre du projet d'Endoscope Virtuel en effet, le prototype actuel ne gère que la conversion au format QuickTime, le contenu de ce chapitre pourra permettre, aux futur développeurs, d'utiliser des outils de codages qui lui seraient proposés sur internet en étant déjà familier avec le vocabulaire propre du domaine de la compression vidéo et, de connaître plus précisément la norme MPEG-2, qui se présente comme 'la norme' à utiliser en matière d'imagerie médicale.

# **PARTIE IV Discussion**

## **Introduction**

Dans la partie Objectifs et Choix, nous avons présenté les objectifs que les applications "Interface Commune" et "Microscope Virtuel" devaient rencontrer. A cette fin, nous avons entrepris une série de choix permettant de respecter les buts fixés.

Par la suite, nous avons exposé les solutions informatiques développées, non seulement en explicitant l'architecture interne des applications à l'aide de diagrammes de classes commentés, mais aussi en illustrant à travers une série d'images les résultats obtenus en suivant la modélisation et les choix effectués.

Afin de mieux comprendre les objectifs atteints et les limites des applications, cette partie analysera, pour les deux outils développés, premièrement l'adéquation des résultats obtenus avec les résultats attendus (objectifs) ainsi que les limites actuelles et, deuxièmement, les perspectives qui proposeront des pistes à suivre pour les futurs développeurs.

## **10.1 Interface Commune, discussion de la solution**

Voici un rappel des objectifs à atteindre pour ce projet et pour chacun d'eux, une discussion sur ce qui a été atteint ou non.

### **10.1.1 Une interface ergonomique et intuitive**

Il est difficile de se prononcer actuellement sur l'ergonomie de l'interface car, le projet n'étant qu'à son commencement, aucun scientifique excepté Hubert Meurisse, n'a encore pu tester le logiciel. Cependant, l'application permet comme voulu dans les objectifs, une liberté d'utilisation; c'est à dire que l'utilisateur peut travailler à, sa guise, sur un seul ou plusieurs outils d'imageries à la fois.

### **10.1.2 Ajouter et rassembler des fonctionnalités communes**

Comme nous l'avons vu dans les screenshots, chaque outil a accès aux fonctionnalités communes propres à l'utilisation d'un type de fichier( grâce au concept de 'Source'). Les méthodes DICOM ont, quand à elles, été rassemblées dans un outil spécifique: 'Panel Envois'. Cela permet à tout les programmes intégrés d'avoir accès aux données du patient, via l'objet partagé DDOManager et, de les envoyés sur le réseau DICOM avec les fichiers d'imagerie capturés. La cohérence des données est assurée par le fait qu'il n'y a qu'un seul DDOManager. Cet objectif a donc été rempli.

### **10.1.3 Une conception qui permet flexibilité et maintenabilité**

En ce qui concerne la flexibilité, les outils d'imageries, qui étendent la classe Outil, elle-même étendant la classe Jpanel, peuvent être disposés sur une fenêtre JAVA de beaucoup de façons différentes. En effet, on pourrait imaginer un système autre que celui d'onglet proposé, il n'y aurait pas besoin pour autant de modifier les outils, il suffirait de changer la présentation de la fenêtre principale puis de rajouter les outils à l'endroit prévu.

De plus, la conception permet aussi de choisir quels sont les outils que l'on souhaite avoir ou non dans l'interface.

Au niveau de la maintenabilité, le fait que tout les outils actuellement disponibles(par exemple l'outil Base de Donnée) ont été intégrés dans l'interface prouve qu'il est facile de rajouter de

nouveaux outils. Pour les fonctionnalités propre à chaque type de fichier, la conceptualisation en 'Source' et le SourceManager permet à tout programmeurs de rajouter ,soit de nouvelles fonctionnalités à des classes Source déjà existantes soit, de créer de nouvelles classes propres à un nouveau type de fichier qui doit être pris en charge par l'application. Ce troisième objectif a donc lui aussi été atteint.

#### 10.1.4 Remarque

Actuellement, lorsque le programme est quitté d'une autre façon que par celle proposée par l'interface (via à le menu fichier ou l'icône de fermeture de fenêtre), par exemple lors d'un plantage de windows, les fichiers stockés dans le Panier ne sont pas perdu, ils sont sauvegardé dans un dossier 'TEMP'. Cela à été prévu pour assurer un minimum de robustesse à l'application. En cas de plantages hardware ou logiciel, les images capturées par les outils et envoyées dans le Panier peuvent donc être récupérées manuellement dans le dossier de l'application. Il serait intéressant de proposer à l'utilisateur la récupération de ces fichiers dans le Panier lors du démarrage de l'application.

## 10.2 Interface Commune, perspectives

### 10.2.1 La sélection des outils

Même si l'Interface Commune permet de travailler avec un nombre quelconque d'outils, cela se passe pour l'instant au niveau 'programmeur', c'est le développeur lorsqu'il implémente la classe Onglet qui choisit quels sont les outils présents ou non. Il serait plus judicieux de proposer au lancement de l'application une fenêtre offrant à l'utilisateur la possibilité de choisir quels sont les outils qui seront présents dans les onglets. Il est à noté que l'outil Panel Envois doit toujours être présent.

## 10.3 Endoscope Virtuel, discussion de la solution

Rappelons le, l' objectif principal était un prototype de lecteur vidéo, doté de fonctions basiques plus l'enregistrement et la capture d'images. L' objectif étant très vague, nous allons parler de ce que peut faire ou non l'application actuelle.

### 10.3.1 Les Fonctionnalités de bases

#### Lecture

En ce qui concerne la lecture d'un flux provenant de la webcam et du magnétoscope, tout fonctionne correctement. Pour ce qui est de lire des fichiers vidéo, le prototype est limité a ce que JMF peut lire, et cela ne comprend pas tout les types de fichier.

#### Snapshot

La capture d'image au format JPEG se fait comme souhaitée de plus, la qualité des images est celle de la vidéo.

## Enregistrement

Pour ce point, JMF n'a pas tenu toute ses promesses et ,nous ne pouvons enregistrer les vidéos qu'au format quicktime (.mov). De plus, l' enregistrement n'est possible actuellement que pour un flux provenant d'un appareil et non d'un fichier stocké sur disque dur. Cela est dû à l'architecture particulière de JMF qui ne considère pas de la même façon une source vidéo provenant d'un flux de streaming d'une vidéo stockée dans un fichier.

En ce qui concerne la qualité des vidéos enregistrées, cela varie en fonction de l'appareil d'entrée, les fichiers enregistrés avec la webcam sont de qualité identique à ce qui est affiché à l'écran. Pour le magnétoscope, la vidéo enregistrée est de moins bonne qualité dans le sens où il y a des ralentissements sur certaines vidéo.

### 10.3.2 Affichage du flux vidéo

Nous pouvons voir sur les screenshots de l' Endoscope Virtuel dans la partie Résultats que la taille de la fenêtre varie en fonction de l'input, elle est plus grande lorsque le magnétoscope est en entrée (via le système Pinnacle) que lorsque c'est la webcam qui est connectée. Cela est un bon point car l'application s'adapte en fonction du format d'affichage vidéo de l'appareil qui est connecté. Le côté négatif est que la taille des boutons accompagnés par la fenêtre d'affichage( les boutons Snapshot, Stop et record) varient aussi. Il faut donc corriger cette petite erreur de codage.

### 10.3.3 Autres remarques sur l'application actuelle

Il est à noter que la qualité de la vidéo affichée par JMF est dépendante des drivers utilisés pour les périphériques installés sur la machine. Dans les cas du système Pinnacle, les drivers FireWire d'origines de windows se sont révélés beaucoup plus performant que ceux installés avec le logiciel fournit par le constructeur de la carte FireWire .

## 10.4 Endoscope Virtuel, perspectives

### 10.4.1 L 'ajout d'effets visuels

Comme présenté dans les objectifs, il serait intéressant pour l'application d'endoscopie de fournir des outils d'imagerie capables de fournir une aide au diagnostic pour le praticien. La possibilité d'ajouts visuels en temps réel sur la vidéo provenant d'un appareil de capture ou d'un fichier enregistré pourrait s'avérer très utile pour repérer d'éventuelles anomalies ou pour mettre en évidence certains éléments de l'image.

A cette fin, le Java Multimedia Framework permet aux programmeurs à travers les interfaces Codec et Effect .

1) Une classe Codec est une unité qui fournit un travail de calcul sur un média, elle accepte en input un objet du type 'Buffer', transforme les données d'input puis stocke le résultat dans un autre objet du type 'Buffer' et le transmet en output. Elle ne peut avoir qu'un seul input et qu'un seul output. Des exemples typiques de fonctionnalités apportées par une telle classe sont le codage et décodage de flux audio et/ou vidéo, ainsi que l'ajout d'effets visuels dans une vidéo.

2) La classe Effect ressemble à une classe Codec mais, elle fournit uniquement des fonctionnalités d'ajouts d'effets spéciaux sur les données d'entrée. Une autre différence est que cette classe ne modifie pas le format du fichier ou du streaming d'input, elle ne fait que manipuler leur contenu. Voici maintenant deux images qui donnent un exemple de ce que l'on peut faire dans le domaine

des effets spéciaux avec JMF. Les images sont prise du site <http://www.exactfutures.com/> . Ce site propose une série de classes basées sur les bibliothèques JMF qui implémentent des algorithmes de traitements d'images tels que la détection de mouvements ou l'ajout d'informations à l'écran (nombre de frames par minutes, ...).

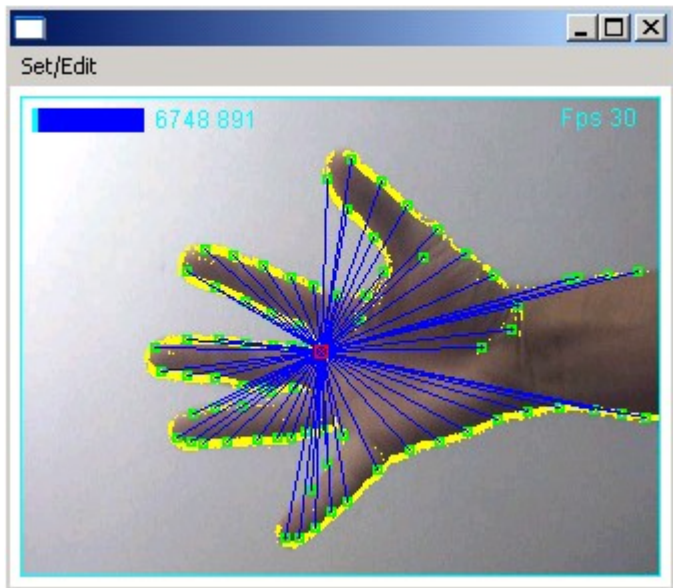


Illustration 31: JMFeffet-exemple1

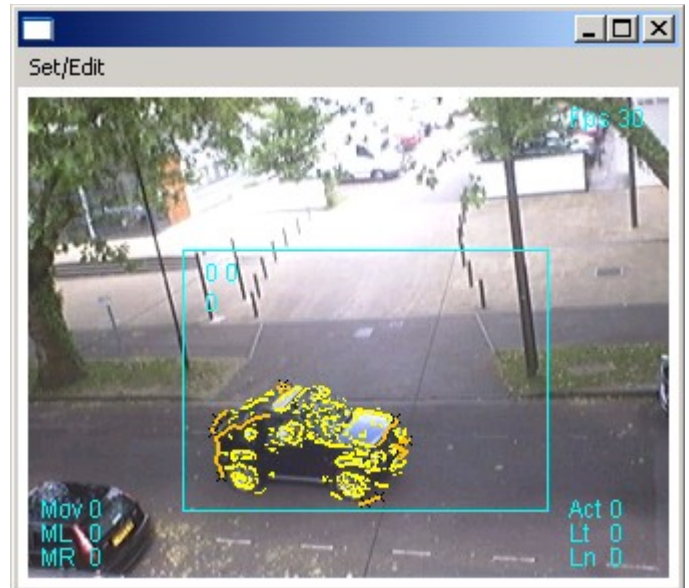


Illustration 30: JMF-effet-exemple 2

#### 10.4.2 Les bibliothèques open-source

Dû aux limites des bibliothèques JMF développées par Sun Microsystems et au manque de mise à jour du package (la dernière mise à jour remonte au mois de novembre 2004), plusieurs projets open-source ont vu le jour avec pour objectif de dépasser les manques actuels de JMF. A cette fin, ces équipes de développement créent leurs propres classes qui soit, sont compatibles avec les classes JMF, soit remplacent certaines classes du package JMF original.

Lors de l'élaboration de ce travail, ces packages open-source ont été testés mais ne présentaient pas de fonctionnalités assez intéressantes pour être utilisés dans les applications. Cependant, ces projets étant en perpétuelle évolution, il est très intéressant de suivre de près leur stade de développement et de les intégrer dans les applications dès qu'ils présenteront un intérêt particulier. De plus, si d'autres projets voient le jour, ils devraient être référencés sur le site des projets open-source existant.

Voici deux de ces projets.

##### 1) Freedom for Media in Java (FMJ)

FMJ est un projet open-source avec pour objectif de fournir une alternative à JMF. Il a pour but de produire des bibliothèques permettant la capture, l'affichage, l'ajout d'effets spéciaux sur des médias à partir de n'importe quel système d'exploitation.

Voici le site internet du projet: <http://fmj.sourceforge.net/>



## 2) FOBS4JMF

FOBS est un projet qui fournit des librairies JAVA qui rendent les classes de JMF capables de communiquer avec les libraires FFMPEG. FFMPEG est un autre package écrit dans le langage C dédié au traitement des flux numériques (enregistrement, lecture ou conversion d'un format à un autre).

Voici le site internet de FOBS : <http://fobs.sourceforge.net/>

# PARTIE V CONCLUSION

Les besoins informatiques d'un hôpital tel que celui de Mont-Godinne sont en constante évolution et les applications nécessaires peuvent recouvrir plusieurs domaines, la télé-médecine, la communication inter-systèmes ou encore, les techniques d'imagerie médicale.

Afin de répondre aux besoins particuliers des différents départements de médecine, le service de recherche informatique développe en interne des applications qui, pour certaines, n'ont pas d'équivalent sur le marché. Ces projets sont menés en partie par des mémorants lors de leur stage de fin d'études.

Ce mémoire basé sur le stage 2006-2007 effectué à l'hôpital de Mont-Godinne s'est focalisé sur les difficultés rencontrées par le service d'endoscopie en particulier. En proposant un outil informatique appelé Endoscope Virtuel, ce travail démarre un projet qui, à long terme, permettra aux scientifiques d'accéder à des fonctionnalités non accessibles par les logiciels actuels. De plus, l'application leur permettra une gestion plus aisée des fichiers d'imageries acquis lors de l'étude d'un patient et donc un travail au quotidien plus facile et plus efficace.

Un autre projet nommé Interface Commune à aussi vu le jour cette année 2006-2007, cette application fait suite à la problématique posée par la pluralité des outils d'imageries et à la dispersions de leurs fonctionnalités.

A fin de mener à bien ces deux projets, nous avons dans un premier temps posé les objectifs à atteindre pour cette année 2006-2007 puis, nous avons effectué une série de choix qui ont permis d'implémenter une solution qui soit en adéquation avec les attentes des scientifiques.

Le résultat est d'une part une application polyvalente orientée-objet qui permet d'acquérir et de diffuser à l'écran la vidéo provenant d'un quelconque appareil connecté à l'ordinateur et reconnu par le système d'exploitation windows. Et d'autre part, une application qui intègre les différents outils d'imageries développés en interne afin de leur fournir des fonctionnalités qu'ils ne disposaient pas jusqu'alors.

Ces deux applications sont sujettes à des améliorations ou modifications futures, c'est pour répondre à ce caractère changeant de l'informatique moderne que la conception de ces deux projets se veut la plus évolutive possible. La partie Discussion a proposé dans ce sens quelques pistes d'évolutions pour les développeurs suivants.

Il est à noter que l'Endoscope Virtuel, étant à son point de démarrage, n'a pour l'instant pas été soumis à la critique des scientifiques excepté Monsieur Hubert Meurisse. Il est important donc à l'avenir d'intégrer les principaux utilisateurs de l'application c'est à dire les scientifiques du département d'endoscopie. C'est en récoltant leurs avis et leurs visions qu'il sera possible pour l'application de répondre aux mieux à leurs exigences et donc d'apporter une valeur ajoutée dans leur travail au quotidien.

# BIBLIOGRAPHIE

[COLV] Medisys, AJR, <http://www.medisys.ca/fr/imagerie-medicale/coloscopie-virtuelle.htm>, janvier 2005.

[DICOM] <http://medical.nema.org/> site officiel du standard DICOM.

[IGE] IMARC R. MAYBERG, M.D., ERIC LAPRESTO, M.S., AND EDWIN J. CUNNINGHAM, M.D. Image-guided endoscopy: description of technique and potential applications, Seattle Neuroscience Institute, Seattle, Washington; and Department of Neurosurgery, Cleveland Clinic Foundation, Cleveland, Ohio Neurosurg. Focus / Volume 19 / July, 2005.

[JADC] Cédric Jadoul Etude et développement d'un système de télé-microscopie virtuelle, Facultés Universitaires Notre-Dame de la Paix, Namur Institut d'Informatique Année Académique 2004 – 2005.

[JMF] <http://java.sun.com/products/java-media/jmf/> site officiel du Java Multimedia Framework Copyright 1994-2007 Sun Microsystems, Inc.

[MPEG] <http://www.chiariglione.org/mpeg/> site officielle du Movie Picture Expert Group.

[MPEGT1] [http://www.zenith.com/sub\\_hdtv/mpeg\\_tutorial/index.htm](http://www.zenith.com/sub_hdtv/mpeg_tutorial/index.htm) Tutorial sur le format MPEG-2 et les techniques de vidéo-compression, his tutorial is copyrighted 1998, 1999, by Wayne Bretl and Mark Fimoff .

[MPEGT2] Victor Lo, City University of Hong Kong , tutorial sur le format mpeg-2, <http://www.fh-friedberg.de/fachbereiche/e2/telekom-labor/zinke/mk/mpeg2beg/beginnzi.htm#Introduction>.

[MPEGT3] Chad Fogg , [http://bmrc.berkeley.edu/research/mpeg/faq/mpeg2-v38/faq\\_v38.html](http://bmrc.berkeley.edu/research/mpeg/faq/mpeg2-v38/faq_v38.html) , tutorial sur le format MPEG et les techniques de compressions vidéo basiques, The Berkeley Multimedia Research Center, université de Californie, Berkeley, (April 2, 1996) .

[PMIA] Tan Chiew Seng Sean<sup>1</sup>, Bertil Schmidt<sup>2</sup> Combining Message-passing and Inter-process communication for Efficient Parallel Medical Image Analysis Biomedical Engineering Research Centre, <sup>1</sup>School of Electronic and Electrical Engineering, <sup>2</sup>School of Computer Engineering, Nanyang Technological University, Singapore 639798.

[WIK] [http://fr.wikipedia.org/wiki/Accueil\\_Wikimedia\\_fundaton\\_inc](http://fr.wikipedia.org/wiki/Accueil_Wikimedia_fundaton_inc). Encyclopédie .

# Annexe I Matériel de test

Cette annexe présente les fiches techniques des deux appareils utilisés pour les tests vidéo avec l'outil Endoscope Virtuel.

## 1) Le système Pinnacle



### Général

Type de périphérique: Adaptateur d'entrée vidéo

Type de châssis: Carte enfichable

Type d'interface: PCI

Compatibilité: PC

### Vidéo

Type: Carte d'acquisition vidéo

Facteur de forme: Carte enfichable

Type d'interface: IEEE 1394 (FireWire)

Format de vidéo analogique: PAL

Format de vidéo numérique: MPEG-1, MPEG-2, AVI, DV, RealVideo 8

Fréquence de trame: PAL : 25 trames par seconde

Résolution de la capture vidéo numérique: 720 x 576

Format de photo numérique: JPEG, BMP, TIFF, TGA, PICT, WMF

Support d'entrée audio: Standard

## **Extension/connectivité**

Interfaces:

- 2 x IEEE 1394 (FireWire) - FireWire 6 broches - externe
- 1 x IEEE 1394 (FireWire) - FireWire 6 broches - interne
- 1 x IEEE 1394 (FireWire) - FireWire 4 broches

Connecteurs compatibles: 1 x PCI

## **Logiciels / configuration requise**

Logiciel inclus: Pinnacle Studio 9

Système d'exploitation requis: Windows 98SE/2000/ME/XP

Périphériques / interfaces: Carte son, CD-ROM, carte graphique, connecteur PCI

Configuration requise: Windows 98SE/2000/ME/XP - 800 MHz - RAM 256 Mo - DD 500 Mo

## **2) La webcam**



## **caractéristiques technique**

- Caméra Web avec capteur CMOS CiF de qualité (352 x 288 pixels)
- Capture vidéo: 640 x 480 pixels (interpolation logicielle)
- Capture photo: 640 x 480 pixels (interpolation logicielle)
- Débit d'images: jusqu'à 30 images par seconde (sur les systèmes possédant la configuration recommandée)
- Casque fourni

- Certifié USB 2.0
- Câble USB de 2 m
- Mise au point manuelle
- Clip universel pour écrans LCD, CRT ou ordinateurs portables
- Garantie de 2 ans

### **Configuration requise**

- Windows® 2000 ou XP
- Processeur Pentium® III 1 Ghz, Celeron®, AMD Athlon® ou supérieur
- 128 Mo de RAM (ou la configuration minimale requise par le système d'exploitation, selon la quantité la plus élevée)
- 200 Mo d'espace libre sur le disque dur
- Port USB 1.1 ou 2.0 disponible
- Lecteur de CD-ROM
- Carte son compatible Windows® et haut-parleurs (carte son duplex intégral recommandée)

## **Annexe II JMF, informations complémentaires**

Cette annexe propose le matériel hardware et software afin de pouvoir utiliser correctement les librairies de JMF. Il s'en suit un tableau récapitulatif sur les appareils d'entrée reconnu par JMF.

### **1) Configuration requise**

#### **Exigences hardware**

- 486 or Pentium based PC (Pentium 100MHz ou meilleur).
- Carte vidéo supportant les modes d'affichage 8, 16, 24 ou 32 bit . ( mode d'affichage 16-bit recommandé pour une performance vidéo optimale).
- Option: une carte audio SoundBlaster-compatible pour les machines windows sans carte audio d'origine.

#### **Exigences logicielles**

- Windows 95/98 or Windows NT 4.0.
- JDK 1.1.3 ou version ultérieure pour Windows . JDK 1.1.6 ou version ultérieure est requi pour une compatibilité Y2K.
- Microsoft DirectShow(recui pour les playback MPEG).
- Les classes JMF et les librairies natives (inclus dans JMF 2.1.1).
- Optionel: Les exemples d'applets et d'applications (inclus dans JMF 2.1.1).
- Optionnel: Microsoft DirectX 2.1 ou ultérieur
- Optionnel: HotJava Browser 1.1.x (deSun).
- Optionnel: Netscape Communicator 4.03-4.05 avec le patch JDK 1.1 ou Netscape Communicator 4.06 ou supérieur
- Optionnel: Microsoft Internet Explorer 4.0 ou supérieur; 3.02 (nécessite Java Plug-in).

**2)Voici un tableau récapitulant les appareils de capture supportés par JMF 2.1.1.**

<b>Capturer</b>	<b>JMF 2.1.1 Cross Platform Version</b>	<b>JMF 2.1.1 Solaris Performance Pack</b>	<b>JMF 2.1.1 Windows Performance Pack</b>
JavaSound (16-bit, 44100, 22050, 11025Hz, 8000Hz linear)	<b>X (J2SE 1.3+)</b>	<b>X</b>	<b>X</b>
SunVideo	-	X	-
SunVideoPlus	-	X	-
VFW	-	-	X
Intel Create & Share	-	-	Win9x
Diamond Supra Video Kit; Share	-	-	Win98
QuickCam VC (camera)	-	-	WinNT
e-cam (camera)	-	-	WinNT, 9X
Winnow Videum	-	-	WinNT, 9X
Creative Web Cam II	-	-	Win9X
Miro Video DC30	-	-	Win9X
Iomega Buz	-	-	Win9X
QuickCam Home USB (Camera)	-	-	Win98
Smart Video Recorder III	-	-	Win9X